Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

2010-09

Behavioral analysis of network flow traffic

Heller, Mark D.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/5108

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

BEHAVIORAL ANALYSIS OF NETWORK FLOW TRAFFIC

by

Derby C. Luckie
and
Mark D. Heller

September 2010

Thesis Co-Advisors:	Geoffrey G. Xie John Gibson Michael Collins
Second Reader:	Raymond Buettner

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2010	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Behavior Analysis of Network Flow Traffic	5. FUNDING NUMBERS
6. AUTHOR(S) Derby C. Luckie and Mark D. Heller	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Information Systems Agency (DISA), PEO-IA22 PO Box 4502 Arlington, VA 22204	10. SPONSORING/MONITORING AGENCY REPORT NUMBER PMA-10-010

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: ___N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Network Behavior Analysis (NBA) is a technique to enhance network security by passively monitoring aggregate traffic patterns and noting unusual action or departures from normal operations. The analysis is typically performed offline, due to the huge volume of input data, in contrast to conventional intrusion prevention solutions based on deep packet inspection, signature detection, and real-time blocking. After establishing a benchmark for normal traffic, an NBA program monitors network activity and flags unknown, new, or unusual patterns that might indicate the presence of a potential threat. NBA also monitors and records trends in bandwidth and protocol use.

Computer users in the Department of Defense (DoD) operational networks may use Hypertext Transport Protocol (HTTP) to stream video from multimedia sites like youtube.com, myspace.com, mtv.com, and blackplanet.com. Such streaming may hog bandwidth, a grave concern, given that increasing amounts of operational data are exchanged over the Global Information Grid, and introduce malicious viruses inadvertently. This thesis develops an NBA solution to identify and estimate the bandwidth usage of HTTP streaming video traffic entirely from flow records such as Cisco's NetFlow data.

14. SUBJECT TERMS NetFlow, flow, NIPRNet, traffic monitoring, bandwidth, DISA, Centaur, SiLK, Network Behavior Analysis, and flow analysis.	15. NUMBER OF PAGES 95
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

BEHAVIORAL ANALYSIS OF NETWORK FLOW TRAFFIC

Derby C. Luckie
Lieutenant Commander, United States Navy
B.S., University of Maryland, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Mark D. Heller
Lieutenant, United States Navy
B.S., United States Naval Academy, 2002

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN COMPUTER SCIENCE
and
MASTER OF SCIENCE IN INFORMATION WARFARE SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2010

Authors:	Derby C. Luckie	Mark D. Heller
Approved by:	Geoffrey G. Xie Co-Advisor	John Gibson Co-Advisor
	Michael Collins Co-Advisor	Raymond Buettner Second Reader
	Peter Denning Chairman Department of Computer Sciences	Dan Boger Chairman Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Network Behavior Analysis (NBA) is a technique to enhance network security by passively monitoring aggregate traffic patterns and noting unusual action or departures from normal operations. The analysis is typically performed offline, due to the huge volume of input data, in contrast to conventional intrusion prevention solutions based on deep packet inspection, signature detection, and real-time blocking. After establishing a benchmark for normal traffic, an NBA program monitors network activity and flags unknown, new, or unusual patterns that might indicate the presence of a potential threat. NBA also monitors and records trends in bandwidth and protocol use.

Computer users in the Department of Defense (DoD) operational networks may use Hypertext Transport Protocol (HTTP) to stream video from multimedia sites like youtube.com, myspace.com, mtv.com, and blackplanet.com. Such streaming may hog bandwidth, a grave concern, given that increasing amounts of operational data are exchanged over the Global Information Grid, and introduce malicious viruses inadvertently. This thesis develops an NBA solution to identify and estimate the bandwidth usage of HTTP streaming video traffic entirely from flow records such as Cisco's NetFlow data.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
A. BACKGROUND 1
1. Web Caching 1
2. Multimedia Streaming 1
3. Network Behavior Analysis 2
B. THESIS OBJECTIVE 3
C. DEFENSE INFORMATION SYSTEMS AGENCY 3
D. CENTAUR 4
E. NETFLOW 4
F. ORGANIZATION 5

II. DOD NIPRNET ARCHITECTURE 7
A. GLOBAL INFORMATION GRID 7
B. GIG COMPONENTS 7
1. Hardware 8
2. Data 8
3. User 8
C. DEFENSE INFORMATION SYSTEM NETWORK (DISN) 9
1. DISN Information Transfer Facilities 10
a. NIPRNet 10
b. SIPRNet 11
D. SUMMARY 12

III. NIPRNET INAPPROPRIATION 13
A. NETWORK AVAILABILITY 13
B. INTERNET EVOLUTION 14
C. NIPRNET GOVERNANCE 16
D. RECREATIONAL INTERNET USE 17

IV. EXPERIMENTATION AND EVALUATION ON CONTROLLED DATASETS 21
A. DATA COLLECTION 22
1. Web site Selection 23
2. Collection Process 24
B. ANALYSIS 25
1. Classification by IP Address 25
a. Proxy Server 27
b. Content Distribution Networks 28
c. Limitations of IP Based Classification 30
2. Classification by Keywords 32
a. Limitations 36
3. Classification by IO Plots 37
4. Asymmetric Transfer Characteristics 40

a. Limitations of Asymmetric Transfer Characteristics 42
5. SiLK 42
C. SUMMARY 44
V. CENTAUR ANALYSIS 45
A. FLOWS 45
B. REPOSITORY QUERY 46
C. VISUAL ANALYSIS 47
D. VERIFICATION 50
E. TRENDS 54
F. SUMMARY 56
VI. CONCLUSION AND RECOMMENDATIONS 59
A. STUDY OVERVIEW 60
B. FUTURE WORK 61
C. RECOMMENDATIONS 62
APPENDIX 63
A. KEYWORDS.PL 63
B. SILK_PARSER.PL 66
C. REVERSEDNS.PL 69
D. LIST OF KNOWN OR SUSPECTED CDN NETWORKS 70
LIST OF REFERENCES 75
INITIAL DISTRIBUTION LIST 77

LIST OF FIGURES

Figure 1. Multi-server content topology 2
Figure 2. NIPRNet Topology..... 11
Figure 3. DoD Internet Activity from October 2004—July 2010–8, 2007 from DISA..... 18
Figure 4. DoD Internet Web Browsing by Category “Top 25 Web sites” reflected 01–30 June 2010 19
Figure 5. YouTube Activity for Feb 5–8, 2007 FROM “YouTube Activity on NIPRNet From May 2005 to December 2006”..... 20
Figure 6. Typical Proxy Server 28
Figure 7. Typical video streaming under the CDN model 29
Figure 8. Non-streaming browsing where the X-axis is the session duration in seconds and the Y-axis is bits per second. Note the peaks followed by no traffic. 38
Figure 9. Tube8 12 Aug sample where the X-axis is the session duration in seconds and the Y-axis is bits per second. In this figure, the pattern repeats at a constant rate until the stream finishes. 39
Figure 10. 666MB Linux ISO download where the X-axis is the session duration in seconds and the Y-axis is bits per second. This shows the filling of the systems buffer, the TCP back-of, and the resumption of the file transfer. 39
Figure 11. Flows generated. Flow 1 is from A→B and Flow 2 is from B→A. 46
Figure 12. Query Builder for SiLK..... 47
Figure 13. Flow bps of both testing data (red) and captured DISA data (blue >45 second duration and cyan < 45 seconds). The DISA data samples are sorted in decreasing session size while the testing data samples are placed at random positions between 0 and 35000 for visual effect. 49
Figure 14. Flow bps vs sample ID of both testing data (red) and captured DISA data (blue and cyan). The orange curve represents a 20 second minimum duration, the yellow is 45 seconds and the purple 80 seconds. 50
Figure 15. Matching records and CDNs per Matching records. 76,911 Total Records in sample. 53
Figure 16. Unique IPs and CDNs per Unique IPs. 53
Figure 17. Percent matching vs Minimum session size 54
Figure 18. Percent of flows classified as CDN with six different minimum session size thresholds (3, 4, 5, 10, 15, and 20 MB)..... 55
Figure 19. Percent of traffic volume classified as CDN with six different minimum session size thresholds (3, 4, 5, 10, 15, and 20 MB) 55

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Internet Usage and Population Statistics 14
Table 2. Names, rankings and categories of test site provided by Alexa 24
Table 3. Initial data capture 25
Table 4. Determination of video content in data from YouTube 13Aug capture. We verified that the session with largest Sever to Client transfer was the streaming video 26
Table 5. Video Content Servers 27
Table 6. Sections of a flow graph from EN3 dataset. 34
Table 7. Keyword and Unique IP counts generated by “keywords.pl” * unique IPs not counted due to the absence of streaming video 35
Table 8. Upstream and Downstream bytes and duration for selected datasets. Note the high download:upload ratio 41
Table 9. Upstream and Downstream for File transfers 42
Table 10. Rwstats output from SiLK 43
Table 11. Bps per sample. Note the grouping between 0.3Mbps and 1.2Mbps 44

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AMN	Afghan Mission Network
CERT	Computer Emergency Response Team
COCOM	Combatant Commands
CSUMB	California State University Monterey Bay
DISA	Defense Information Systems Agency
DISN	Defense information System Network
DMS	Defense Message System
DNS	Domain Name Service
DoD	Department of Defense
DRM	Digital Rights Management
DRSN	Defense Red Switch Network
DSN	Defense Switched Network
DTM	Directive Type Memorandum
EDT	Eastern Daylight Time
GCCS	Global Command and Control System
GIG	Global Information Grid
GMT	Greenwich Mean Time
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
IANA	Internet Assigned Numbers Authority
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISP	Internet Service Provider
ITSDN	Integrated Tactical Strategic Data Network
JCS	Joint Chiefs of Staff
JOPES	Joint Operational Planning and Execution System
JTF-GNO	Joint Task Force-Global Network Operations

LAN	Local Area Network
MILDEPS	Military Departments
MPLS	Multiprotocol Label Switching
MTU	Maximum Transmission Unit
NBA	Network Behavior Analysis
NCA	National Command Authority
NETFLOW	Network Flows
NIPRNet	Non-Secure Internet Protocol Router Network
NKO	Navy Knowledge Online
NPS	Naval Postgraduate School
OCONUS	Outside Contiguous United States
P2P	Peer-to-peer
PDT	Pacific Daylight Time
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-time Streaming Protocol
SiLK	System for Internet Level Knowledge
SIPRNet	Secure Internet Protocol Router Protocol
SSH	Secure Shell
STEP	Standardized Entry Point
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
VoIP	Voice over IP
WAN	Wide Area Networks
YAF	Yet Another Flowmeter

ACKNOWLEDGMENTS

GOD for providing us with unwavering wisdom and discernment.

We would like to collectively thank our Thesis Advisors and Readers:

Dr. Geoffrey Xie, Dr. Michael Collins, Dr. Ray Buettner, Dr. Dave Ford, and Prof. John H. Gibson for your research assistance and direction in this project. Your scientific prowess, reasoning, and patience were above reproach.

LCDR Derby Luckie would like to thank the following:

CDR Duane Davis, CDR Mike Bilzor, Dr. Man-Tak Shing, Dr. Dennis Volpano, Dr. Craig Martell, and Prof. J.D. Fulp for your inspiration and personal commitment.

Mr Jim Downey (DISA) for your commitment to this project.

Ms. Maricel M. Eddington (CS Education Technician) for your dedication and charisma.

Special thanks are given to my student-colleagues who helped in so many ways.

Finally, words alone cannot express the sincere appreciation to my family: Ozan, Valerie, Amanda, Victoria, Brandon, and Jan for your perseverance, encouragement, and patience throughout this daunting experience.

LT Mark Heller would like to thank the following:

Dr. Ray Buettner for having confidence in this endeavor and assisting me in completing this thesis.

Dr. Dan Boger, thank you for taking the time to listen to a solution to an unexpected problem.

Dr. Geoffrey Xie and Prof John H. Gibson: Thank you for allowing me to join such an exciting project at a time when I was out of ideas.

LCDR Derby Luckie for his hard work and long hours to accomplish this thesis.

To my wife Kori and son Andrew, It is over. Thanks for the sacrifices of your weekends, evenings and all the other missed opportunities during this process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Effective traffic classification to identify usage of application protocols enhances the ability of network administrators to perform network management tasks, such as overall bandwidth management, network anomaly detection, and network misuse identification, in order to defend Department of Defense (DoD) operational networks.

This thesis focuses on classifying Hypertext Transfer Protocol (HTTP) and behaviors that take place within this protocol. Web caching and multimedia streaming are two such applications that influence Web performance. This thesis will analyze the behavior patterns of multimedia streaming and verify its effect on the network.

1. Web Caching

Web caching moves content closer to a user by sorting the content in the user's browser, or a machine called Proxy Server, which acts an intermediary between the user and the original Web server. Data sent to a Web portal often traverse a proxy server first, mostly in unencrypted form. This presents potential risks to the network, as malicious software may reside on those proxy servers to exploit everything sent, including unencrypted logins and passwords.

2. Multimedia Streaming

During the past several years, multimedia content has become increasingly popular. Streaming Multimedia is used for many tasks, including professional as well as personal. Multimedia streaming applications (e.g., iTunes and YouTube) consume large amounts of already limited bandwidth and are sensitive to delays in receiving audio samples and video frames. Multimedia streaming differs from traditional Web content in data format and performance

with regard to reliable delivery of the data. The browser client decodes video frames as they arrive from the server, rather than downloading the content in its entirety before beginning the playback or display.

Figure 1 depicts a complex Web site with multi-server content requirements included as labels or hyperlinks thoughout the web page. Although, the client is initially directed to a main Web site, the client may be re-directed to alternate servers that provide a particular service or content.

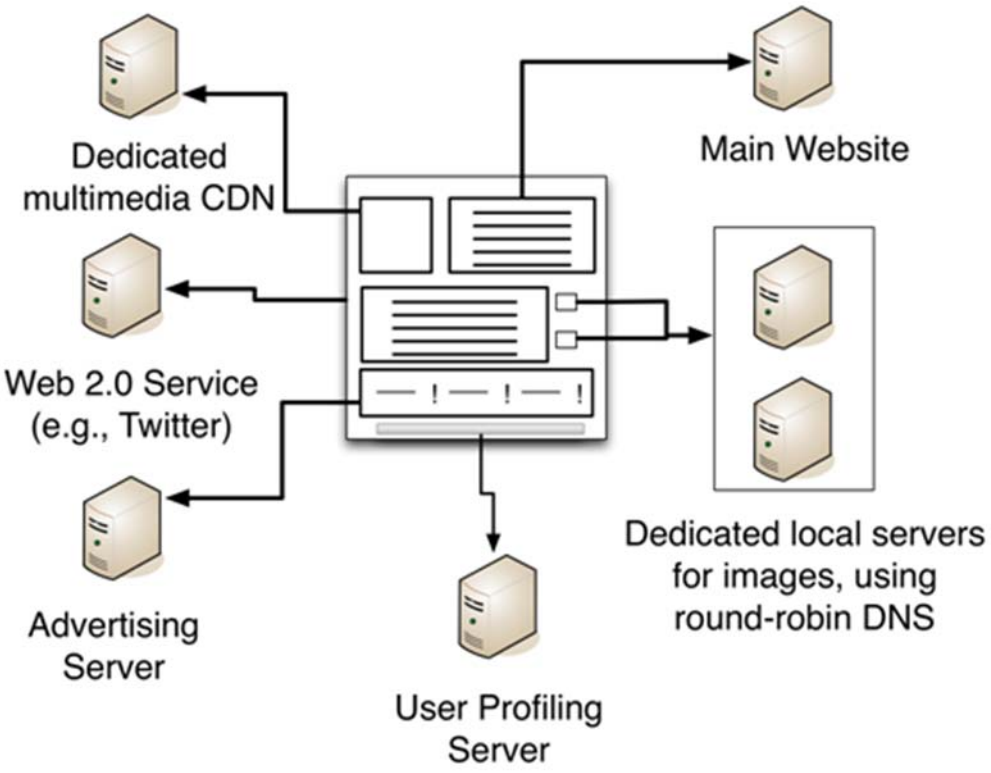

Figure 1. Multi-server content topology

3. Network Behavior Analysis

Network Behavior Analysis (NBA) is a technique to enhance network security by passively monitoring aggregate traffic patterns and noting unusual action or departures from normal operations. The anlaysis is typically performed

offline, due to the huge volume of input data, in contrast to conventional intrusion prevention solutions based on deep packet inspection, signature detection, and real-time blocking. After establishing a benchmark for normal traffic, an NBA program monitors network activity and flags unknown, new, or unusual patterns that might indicate the presence of a potential threat. NBA also monitors and records trends in bandwidth and protocol use.

Network security analysts are often restricted in their ability to analyze traffic behavior patterns because of technological limitations related to querying massive amounts of historical data, as well as the laborious requirement for human interaction to derive answers to complex questions. A critical challenge facing them is discovering new patterns of cyber-adversary behavior and their manifestation as unusual network activity. (Nelson & McAllister 2009) The result of this thesis asserts how we differentiate HTTP streaming video traffic from normal HTTP traffic using flow data collected using the Defense Information System Agency (DISA) Centaur program.

B. THESIS OBJECTIVE

This thesis reports on the authors' actions to discover discriminating signatures for identifying streaming video flows from the Navy's Non-Secure Internet Protocol Router Network (NIPRNet) flow records. Such signatures are obtained using in-place monitoring devices indicating flow start and end time, Internet Protocol (IP) addresses, TCP port numbers, session time, and bytes transferred. Furthermore, this thesis analyzes the utility such signatures to characterize a video traffic profile in sampled NIPRNet data.

C. DEFENSE INFORMATION SYSTEMS AGENCY

Defense Information Systems Agency (DISA) provides global information and technology assistance through online services ("net-centric") for the DoD. DISA helps the United States military forces connect to one another regardless of geographical location, pull information needed for missions, and receive accurate

and protected information on any threats encountered during the process. The agency focuses on speed of delivery of information, operational effectiveness and efficiency, and sharing information. Its primary aim is to provide secure and reliable communications networks, computers, software, databases, applications and other products needed for the processing and transport requirements of the DoD.

D. CENTAUR

Centaur is a DISA created software program consisting of various sensors, storage, and analysis servers used to analyze Cisco NetFlow data. Centaur uses the System for Internet Level Knowledge (SiLK) tool to collect, pack, and query network flow records and provide summarizations for both primary and retrospective analysis (Nelson & McAllister 2009).

The Centaur tools are used to analyze traffic behavior patterns. Typically, analysts must query massive amounts of data to derive an answer to complex questions regarding network behavior.

E. NETFLOW

NetFlow is a *de facto* standard tool used by network analysts and administrators to monitor large enterprise networks. Originally designed by Cisco as a cache to improve IP flow lookups in routers, it soon revealed itself useful for network traffic monitoring and reporting. Implemented by all major vendors and recently an IETF standard, Netflow reports aggregated information about traffic traversing routers in the form of flow records.

A Netflow probe tracks flows, i.e., unidirectional sequences of packets exchanged by two endpoints. First, it extracts from each packet a key composed of specific header fields, i.e., the classical five tuple (IP source and destination addresses, transport layer source and destination ports, and the protocol number). This key identifies a record in memory, where a probe stores, besides the key itself, a number of attributes, like cumulative packets and byte counters,

flow starting and finishing timestamps, IP type of service, TCP flags, Multiprotocol Label Switching (MPLS) label, physical input, and output interface indexes.

Whenever a flow expires, the router transmits a UDP packet containing a NetFlow record to the Netflow collector, which elaborates and eventually stores this information. Different reasons can cause a flow expiration. (1) a packet explicity terminates the flow (e.g., a TCP FIN Flag); (2) the flow has been inactive for a period longer than a presscribed value (default 15 seconds); (3) the flow has been active for a time greater than a prescribed threshold (default 30 minutes); and (4) the flow cache is full and some space needs to be freed for new flows (Rossi and Valenti n.d.).

The use of flow information, specifically with regards to streaming HTTP video traffic classification is central to this study.

F. ORGANIZATION

This thesis presents a hierarchical approach.

Chapter II evaluates and presents current Department of Defense (DoD) SIPRNet and NIPRNet network topology, with emphasis on the Global Information Grid (GIG).

Chapter III discusses effective DoD policies, directives, and instructions used for overall network management.

Chapter IV uses Wireshark to capture and evaluate streaming video obtained to form a known testing dataset. The data is captured from local Internet Service Providers (ISP) and Naval Postgraduate School (NPS) network to develop search profiles. This data is parsed to determine which is the streaming session and explored for possible HTTP streaming video signatures.

Chapter V tests our profiles on the DISA Centaur database using remote traffic analysis tools (e.g., SiLK), to assist in accurately predicting HTTP

streaming video traffic over the DoD NIPRNet. The profile discovered in Chapter IV is tested and evaluated against flow sessions collected from the DISA Centaur database.

Chapter VI compares the effectiveness of the developed method used to determine our profile and the results obtained from the DISA data. The DISA results are analyzed and conclusions generated. Follow-on work is recommended for advancement of this topic.

II. DOD NIPRNET ARCHITECTURE

A. GLOBAL INFORMATION GRID

The Global Information Grid (GIG) must be able to provide accurate, secure, and timely information to commanders anywhere, anytime, and in a format compatible with specific information application requirements. (Saterthwaite 2007). This form of expectation requires a deeper understanding of each information platform to enable seamless integration. The GIG does not just cover a specific network or operational system, but instead provides a layered framework to facilitate communication between multiple functions and various communication protocols.

The GIG vision is to be completely net-centric and operating in a global context, providing processing, storage, management, and transport of information to support all DoD, national security, and related Intelligence Community missions and functions—strategic, operational, tactical, and business in war, in crisis, and in peace. GIG capabilities will be available from all operating locations: bases, posts, stations, facilities, mobile platforms, and deployed sites and will interface with allied, coalition, and non-GIG systems. The overarching objective of the GIG vision is to provide the National Command Authority (NCA), warfighters, DoD personnel, Intelligence Community, businesses, policy makers, and non-DoD users with information superiority, decision superiority, and full-spectrum dominance. The primary goal of the GIG is to provide seamless, protected, reliable, worldwide connection to support mission needs (National Security Agency 2008).

B. GIG COMPONENTS

The GIG consists of three major components: hardware, data, and users. Hardware is the physical system and incorporates technology too detailed to mention in this thesis; however, a short overview follows.

1. Hardware

This technology is divided into four layers: surface, aerospace, near-space, and satellite. The surface layer includes both fixed and mobile communications from actively moving troops, aircraft, or maritime craft. The aerospace layer consists mainly of aircraft (e.g., helicopters, cargo planes, fighters) and traditionally used for intelligence, surveillance, and reconnaissance. The near space layer includes devices piloted from remote locations similar to unmanned aerial vehicles (UAV). Finally, the satellite layer is essential to seamless functioning of the GIG and encompasses a wide range of developed and developing technologies, thus providing "the military with voice and data protected communications capabilities (Satterthwaite 2000).

2. Data

Due to various layers within GIG architecture and the enormous array of technology, measuring the health of GIG becomes quite challenging. The challenge becomes even more difficult concerning aspects of size, type of data, and user requirements. The GIG Architectural Vision identifies the ability to "fully leverage the power of information and collaboration" as the target vision for the GIG. Therefore, it can be concluded that data is the most important component, in such that it is required for both operations of communications and weapons systems, which demands the most protection and attention. Extensive consideration should also be devoted to who is providing the data and the inherent risks to the system (Department of Defense 2007).

3. User

User traffic is defined as information derived from users or user applications; for example, the HTTP request generated and the reply received when a user clicks on a web link. Control traffic is the information transmitted that is essential to ensuring connectivity between the user and the network. An automated bandwidth management processes is an example of control traffic.

Management traffic is information about the status and performance of the network itself, such as updates about vulnerabilities in the network's infrastructure or security information (Buda et al.,n.d.). These three types of data are vital to successful functioning of the GIG; however, it is necessary to determine what type of user data in particular should be protected as some pieces alone are harmless and others may be threatening to national security if delivered to adversaries. It is crucial to determine how this data should be protected, whether via encryption, firewalls, and anti-virus software, regular system status checks, or a combination of all the above.

It is generally accepted that human error is now the primary cause of successful network exploitation. Examples include bypassing security procedures with the use of USB thumb drives, leaving computers unattended, opening emails from unknown sources, downloading information to personal devices to finish work at home, and social engineering; the list is endless. It is imperative to infuse a cultural change in all users that assures reliability and maximum effectiveness to maintain network security.

The capabilities of these layers combine functionality of all GIG layers to provide resources for data management and Information Assurance to form both common and unique Net-Centric capabilities for business and war-fighter segments of the DoD (Satterthwaite, 2000).

C. DEFENSE INFORMATION SYSTEM NETWORK (DISN)

The GIG is centrally operated, managed, and controlled in support of net-centric operations and Joint Task Force-Global Network Operations (JTF-GNO). DISA is responsible for operating and sustaining the DISN, the enterprise computer centers, enterprise services, and command and control capabilities and services (DISA, n.d.). It is DoD's worldwide enterprise-level telecommunications infrastructure, providing end-to-end information transfer in support of military operations. As a critical portion of the GIG, the DISN furnishes network services

to DoD installations and deployed forces, to include: voice, data, video, messaging, and other unified capabilities along with ancillary enterprise services. There are three critical segments of the DISN:

1. Sustaining base: The sustaining base infrastructure (i.e., base, camp, and individual service enterprise enclaves) interfaces with the long haul infrastructure in support of strategic/fixed environment user telecommunications requirements. The sustaining base segment is primarily the responsibility of the combatant commander or specific service.

2. Long haul: The long-haul telecommunications infrastructure and its associated services are the responsibility of the DISA.

3. Deployed: Deployed warfighters and their associated combatant commander telecommunications infrastructures support the Joint Task Forces and/or Combined Task Forces. The combatant command and subordinate Service components have primary responsibility for deployed war-fighters within their theater.(Chairman Joint Chiefs of Staff 2008)

1. DISN Information Transfer Facilities

DISN information transfer facilities support secure transport requirements for sub-networks such as the Defense Switched Network (DSN), Defense Red Switch Network (DRSN), Non-Secure Internet Protocol Router Network (NIPRNet), and Secure Internet Protocol Router Protocol (SIPRNet) (DISA 2010)

a. NIPRNet

The unclassified but sensitive NIPRNet (U.S. Department of Defense 2007) is an IP-based router network for global long-haul network to support unclassified data communications services for combat support applications for the DoD, Joint Chiefs of Staff (JCS), Military Departments (MILDEPS), and Combatant Commands (COCOM). It provides seamless, interoperable, common-user IP services to customers with access data rates ranging from 56kbps to 2.0Gbps via direct connections to a NIPRNet router and

services to the tactical community via Integrated Tactical Strategic Data Network (ITSDN) and Standardized Entry Point (STEP) sites. See Figure 2 for current NIPRNet topology.

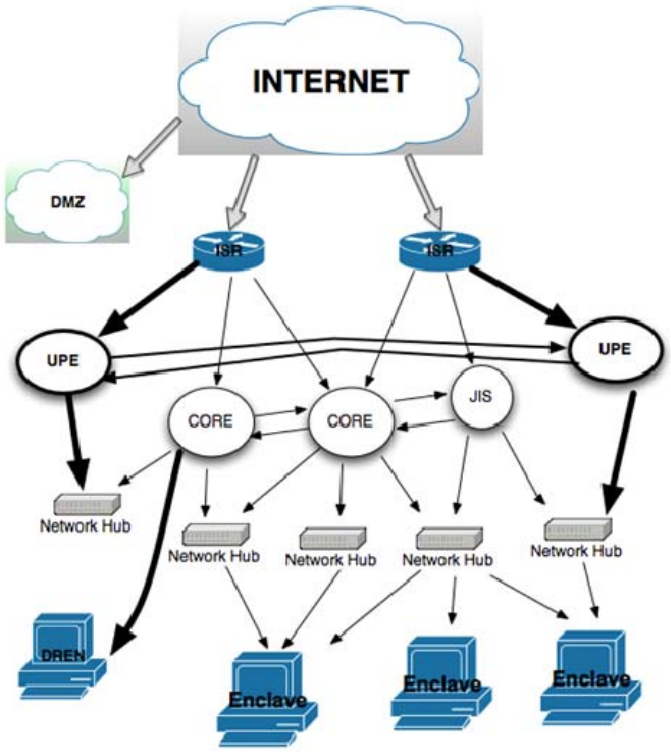

Figure 2. NIPRNet Topology

b. SIPRNet

The SIPRNet is a secret-high IP based router network and is DoD's largest interoperable command and control (C2) data network, supporting the Global Command and Control System (GCCS), the Defense Message System (DMS), collaborative planning through the Joint Operational Planning and Execution System (JOPES), and numerous other classified warfighter applications. SIPRNet provides secure, seamless, interoperable, common-user, packet-switched data communications services to customers with access data

rates ranging from 56 Kbps to 1.0 Gbps. Remote dial-up services are available at up to 115 Kbps, and services to the tactical community are available via ITSDN/STEP sites.

D. SUMMARY

This chapter focused on the overall importance of DoD enterprise networks, as well as their overall management. The enterprise is vitally important to our strategic mission and will continue to be a high priority target continuously under attack from both insider and outsider threats. Degradations, outages, misuse, and cyber attacks will continue and become more prevalent throughout the DoD. Analysts must continue to identify network misuse, as well as cyber attacks to mitigate proliferation to other critical information systems.

III. NIPRNET INAPPROPRIATION

A. NETWORK AVAILABILITY

To ensure operational networks are available for combat operations and their supporting activities, the DoD issued directives for Web site blocking, to prevent DoD computers from accessing specific recreational web sites (e.g., youtube.com, dailymotion.com, and mtv.com). On May 14, 2007, the DoD issued a mandate to block access to 13 Web sites. The DoD directive was a proactive measure to preserve military bandwidth for operational missions and enhance DoD network security. DISA analysts continue to characterize traffic moving across NIPRNet with the caveat that additional sites may be blocked (U.S. Department of Defense 2007).

> Cost is an important thing. Bandwidth. You know when the highest bandwidth rate [utilization] goes up on the NIPRNET? We track this stuff. It's during March Madness. People watching streaming videos of basketball games, checking the scores, checking how their teams are doing in the various brackets. Let me tell you, if you're in the business of making money and you have employees who are spending a couple of hours a day on the payroll doing that kind of work you take that pretty seriously. If you're in the business of fighting our nation's wars and planning for our nation's war and defending this country and you've got people in their spaces and their cubicles doing that, we ought to take that pretty seriously as well. One, from a wasted effort perspective; but other, from opening vulnerabilities. And oh by the way, that bandwidth to see all that stuff? It isn't free. We're paying for that bandwidth. We're leasing that bandwidth often times through commercial capability. (Chilton, 2008)

The most pervasive military networks are the NIPRNet and the SIPRNet. The architecture of these networks parallels that of the Internet, with a large percentage of the unclassified NIPRNet traffic routed through the civilian Internet. Classified SIPRNet messages are logically isolated from the civilian Internet via end-to-end encryption (Dorobek, 2002). Governance of these networks includes

setting clear expectations for network behaviors and actions for all uniformed services and strongly influencing the individuals who manage them to achieve the overall expectations. Governance relies on well-informed decision-making and the assurance that such decisions are enacted as intended with desired outcomes. Enforcing corporate standards and policies requires the means of observing network behaviors and measuring conformance. An essential tool for measuring compliance with standards of conduct for Internet usage is the means to characterize the traffic being generated by the organizations users.

B. INTERNET EVOLUTION

The Internet is formed by a global interconnection of millions of otherwise independent computers, communication entities, and information systems. Millions of people conduct business over the Internet, and millions more use it for personal entertainment. Internet growth has been significant over the past 10 years as seen in Table 1. Connections are growing exponentially; the Internet is adding new networks about every 30 minutes. Because the Internet is a seamless web of networks, it is virtually impossible today to distinguish where one network ends and another begins.

WORLD INTERNET USAGE AND POPULATION STATISTICS

World Regions	Population (2010 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2010
Africa	1,013,779,050	4,514,400	110,931,700	10.9 %	2,357.3 %
Asia	3,834,792,852	114,304,000	825,094,396	21.5 %	621.8 %
Europe	813,319,511	105,096,093	475,069,448	58.4 %	352.0 %
Middle East	212,336,924	3,284,800	63,240,946	29.8 %	1,825.3 %
North America	344,124,450	108,096,800	266,224,500	77.4 %	146.3 %
Latin America/Caribbean	592,556,972	18,068,919	204,689,836	34.5 %	1,032.8 %
Oceania / Australia	34,700,201	7,620,480	21,263,990	61.3 %	179.0 %
WORLD TOTAL	6,845,609,960	360,985,492	1,966,514,816	28.7 %	444.8 %

Table 1. Internet Usage and Population Statistics

What makes this interconnection possible is the set of communications standards, protocol procedures, and formats in common among disparate networks and the various devices connected to them. This infrastructure is constantly evolving to include new capabilities. The protocols initially used by the Internet are called “TCP/IP” and are named after the two protocols that formed the principle basis for heterogeneous inter-network operation. The Internet is a design architecture, although many people confuse it with its implementation. When the Internet is viewed as such, it manifests two different abstractions. One abstraction deals with communications connectivity, packet delivery and a variety of end-to-end communications services. Leveraging this infrastructure is an expanding set of architectural concepts and data structures for disparate information systems that render the Internet truly a global information system. Internet connection enables inter-organizational exchange of electronic mail, logging-on to remote computer sites, downloading and uploading files, as well as streaming video via Content Delivery Networks (CDN). The other abstraction views the Internet as an information system, independent of its underlying communications infrastructure. This “information system” allows creation, storage, and access to a wide range of information resources, including digital objects and related services.

Though clearly beneficial, the Internet also poses serious computer security concerns for the DoD, as well as other government and commercial organizations. The DoD will continue to increase its reliance on the Internet, as it provides early warnings of significant developments quicker than more traditional indications and warnings obtained through normal intelligence channels. Such intelligence provides early indication of activities, whether malicious or unintentional, that would put DoD entities’ access to vital information at risk. Information Dominance, a crucial state in the age of Cyber and Information Warfare, demands that network resources be effectively employed to develop, preserve, focus, and deliver combat power. Increasingly, attempted break-ins and intrusions into government systems are detected daily and these numbers

will continue to rise as information is exchanged over the Internet. To that end, the DoD must protect our information systems, and the infrastructure facilitating these systems.

C. NIPRNET GOVERNANCE

NIPRNet governance represents a series of mandates, constructive initiatives, and activities that are collectively and consensually conceived by governments, public and private sectors, and civil-society organizations. This governance establishes a global-regulation structure that independently promotes scientific, territorial, economic, and social development of the Internet among other nations.

The Assistant Secretary of Defense (ASD) for Command, Control, Communications, and Intelligence (C4I) is designated as lead-agent for DoD systems, to include computer networks. DoD policy requires all systems and networks be monitored in accordance with 18 U.S.C. 2511 and DoD Directive 4640.6 in order to detect, isolate, and react to intrusions, disruption of services, or other incidents threatening security or function of operations, information systems or computer networks. (U.S. Department of Defense n.d.). Directive Type Memorandum (DTM) 09-026 outlines the policies for "Responsible and Effective Use of Internet Based Capabilities." (U.S. Department of Defense n.d.) Access to all publicly accessible information capabilities and applications available across the Internet in locations not owned, operated, or controlled by DoD or Federal Government, by DoD personnel are mandated by this policy. Direct-Type Memorandum 09-026 outlines specific Internet-based capabilities to include MySpace, Twitter, Facebook, MTV and Live365. Each DoD and government entity is responsible for strict adherence to these policies for their respective component.

D. RECREATIONAL INTERNET USE

The DoD's decision to "block" access to certain sites is designed to limit wholesale access to recreational web sites in order to preserve bandwidth needed for mission critical operations. The risk of some sites is commonly known and commentators caution enterprise managers on the disproportionate consumption of network resources by recreational web use. Filtering is the most cost-efficient and time-responsive tool to help ensure the GIG is available and secure to support the war-fighter and their mission requirements. The content on many of the recreational sites is unregulated, user-contributed, and constantly changing. This requires additional resources to defend against access to variable content. This dynamic nature alone facilitates crackers indirectly targeting multiple networks by implanting malicious code within the content and potentially infecting more networks relative to direct targeting. The loss of a critical operational node could have a serious impact on an ongoing military operation. Although the DoD may not be the target of this type of malicious activity, the end result to their networks remain the same.

The potential for network saturation, combined with indicators of increased recreational Internet use have compelled DoD to exercise focused custodial responsibility over its information resources, addressing first the steady rise in commercial Internet access by GIG users. In order to preserve throughput and control the growth curve in overall demand, the DoD began examining ways to limit the impact of inherently recreational Internet activity without impeding legitimate, mission-related browsing (U.S. Department of Defense 2007).

Figure 3 shows the DoD Internet consumption growth, as well as indications that Internet growth has increased at a 39% annual rate. Figure 4 depicts the DoD's Internet Web Browsing "Top 25 Web sites." In addition, a study was conducted by Dr. Michael Collins and sponsored by the CERT

Network Situational Awareness Group examining overall bandwidth usage on the NIPRNet. (Collins 2007) The team observed YouTube usage consuming significant amounts of the bandwidth capacity.

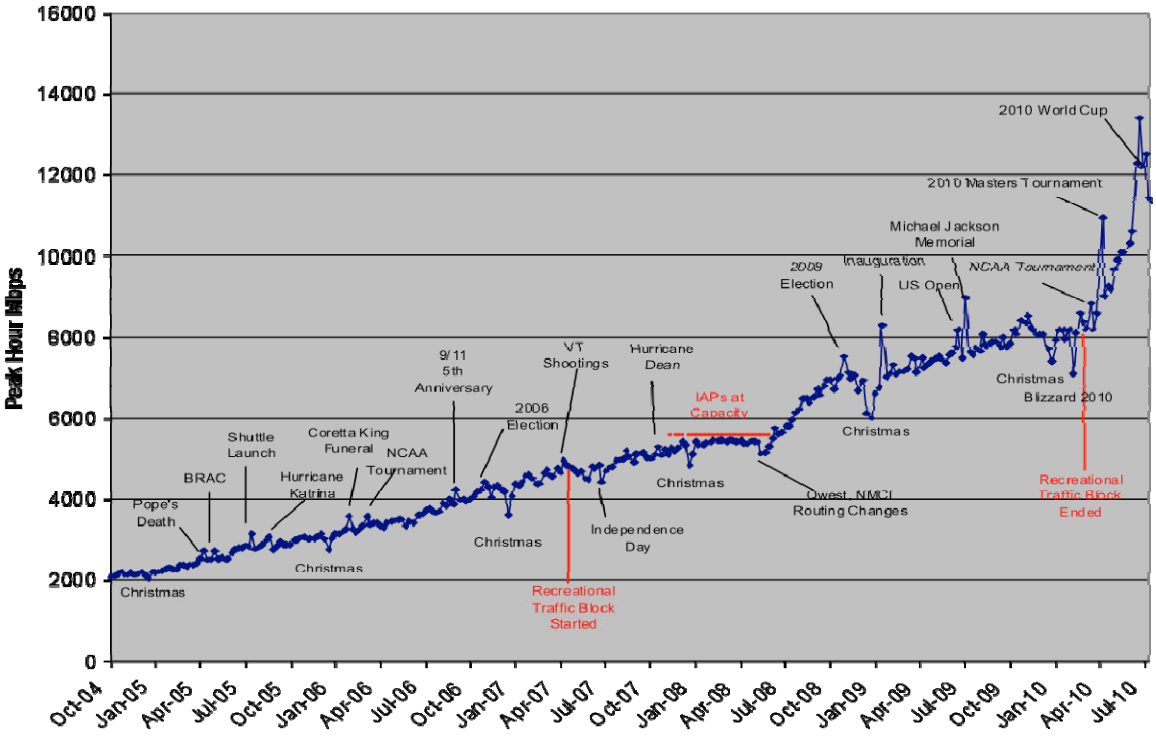

Figure 3. DoD Internet Activity from October 2004—July 2010–8, 2007 from DISA

		Domain	Mbps	%	Description
↑13%	1	google.com (High BW)	2105	15.7%	High-Bandwidth Services including YouTube and Google Video
	2	google.com (Low BW)	713	5.3%	Low-Bandwidth Services including Search, Email and Maps
	3	facebook.com	325	2.4%	Social Networking
	4	yahoo.com	262	2.0%	Search Engine, Portal, News, Personal E-Mail
↑38%	5	youtube.com	236	1.8%	Online Video
	6	streamtheworld.com	218	1.6%	Streaming Radio (Including CBS Radio)
↑44%	7	googlevideo.com	208	1.6%	Online Video
↑	8	espn3.com	183	1.4%	Streaming Sports (Including World Cup)
	9	msn.com	131	1.0%	Portal, News
	10	microsoft.com	129	1.0%	Software and Software Updates
	11	foxnews.com	117	0.9%	News
	12	amazon.com	77	0.6%	Shopping
	13	windowsupdate.com	70	0.5%	Software Updates
↓25%	14	pandora.com	69	0.5%	Internet Radio
	15	symanteclive update.com	59	0.4%	Anti-Virus Software
↑27%	16	espn.go.com	59	0.4%	Sports
	17	ebay.com	51	0.4%	Online Auctions, Shopping
	18	cnn.com	47	0.3%	News
	19	msnbc.com	46	0.3%	News
↑20%	20	af.mil	46	0.3%	Air Force Public Website
	21	doubleclick.com	46	0.3%	Embedded Advertisements
	22	verisign.com	45	0.3%	PKI and Encryption
	23	turner.com	40	0.3%	Multimedia for Turner stations including CNN
	24	eyewonder.com	32	0.2%	Banner Advertisements
	25	newegg.com	31	0.2%	Shopping

Figure 4. DoD Internet Web Browsing by Category "Top 25 Web sites" reflected 01–30 June 2010

By examining YouTube associated flows in a Centaur dataset, it was determined that:

- Peak YouTube downloading to NIPRNet consumes approximately two OC-3 connections of bandwidth (approximately 311 Mbps).
- There is a widespread uploading of videos to YouTube across NIPRNet.
- Bandwidth usage associated with YouTube may be leveling off, which suggests YouTube may have reached full adoption across NIPRNet.

The pattern of growth of YouTube usage on the NIPRNet mirrors growth for video streaming on the Internet at large. Experts predict the NIPRNet users will attempt the same methods of circumventing the YouTube ban as users on corporate networks. The study also noted the amount of traffic downloaded from

YouTube at peak hours (Approximately 130 GB/hr) as seen in Figure 5, is close to the value observed in November 2006. It also indicates that YouTube traffic is a significant fraction (approximately 8%) of all incoming NIPRNet HTTP traffic. DoD users will likely migrate to other competing shared video sites, particularly for unauthorized uploading, as new video sharing Web sites emerge.

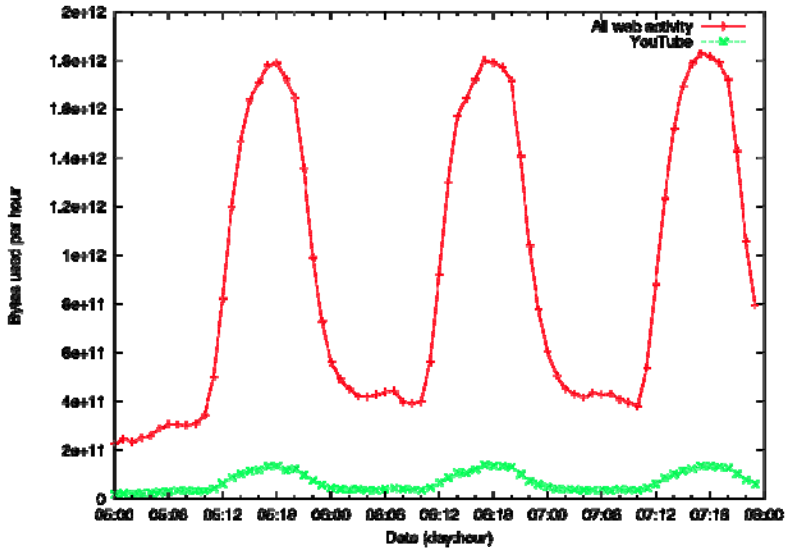

Figure 5. YouTube Activity for Feb 5–8, 2007 FROM "YouTube Activity on NIPRNet From May 2005 to December 2006"

E. SUMMARY

This chapter reviewed overall operations of the NIPRNet and it's significance for overall military operations. In addition, it analyzed the emergence of the Internet, as well as some of the popular services that depend on its existence.

The next chapter will use actual test samples to analyze Hypertext Transfer Protocol behaviors, specifically steaming content and categorize them based on phased analysis.

IV. EXPERIMENTATION AND EVALUATION ON CONTROLLED DATASETS

The goal of this thesis is to evaluate network behavior of a specific network protocol (HTTP) using streaming video. HTTP streaming is one of a variety of multimedia streaming technologies, including Real-time Streaming Protocol (RTSP), Real-time Transport Protocol (RTP), and Real-time Transport Control Protocol (RTCP). The main difference between these protocols and HTTP is that these protocols were designed specifically to support streaming data. With the exception of RTCP and HTTP, the other protocols use UDP to deliver content. During our data collection of streaming video, target sites were using HTTP to deliver streaming content.

Streaming video differs from traditional Web content in three key ways. First, video data has timing properties that affect how helper applications receive and play the streams. Second, multimedia streaming requires the server devote a significant amount of bandwidth over an extended period. Third, unlike other data transfers, multi-media streaming applications are resilient to modest data loss (Krishnamurthy and Rexford 2001). Multimedia sessions over HTTP can involve more than one stream. For example, a single multimedia session might consist of an audio stream and a video stream. Although linked by a common notion of time, these streams may employ different encoding and compression techniques.

In this chapter, we produce, collect, and analyze data from known video streaming Web sites. Our goal is to determine several methods for detecting streaming media and to allow us to predict streaming video on DISA monitored networks. In the following sections, we discuss our research by examining full packet captures to identify signatures of streaming video. We test various methods including IP classification, keyword searches, I/O plot comparisons, and

flow data analysis. Each technique is discussed as we progress from using full packet captures to NetFlow data, which is available on most production networks.

A. DATA COLLECTION

Table 2 summarizes the Web sites used for our exploratory data analysis. The datasets in this table consist of pcap-format traces of network traffic during human-driven surfing sessions to targeted Web sites. Table 3 provides the date when the dataset was collected, the *root provider* (discussed below), the number of unique IP addresses observed, and total traffic volume in bytes and packets.

Browsing a single web page may result in multiple HTTP requests sent to a variety of distinct yet dependent servers. For example, a request to the New York Times Web site (http://www.nytimes.com) may involve a request to a dependent site, such as graphics.nytimes.com, or scripts.nytimes.com, each of which provides necessary data for constructing the page. Therefore, referring to a "single Web site" is often disingenuous, as a web page is composed of requests to multiple Web sites. We use the term **root provider** to refer to the server one would intuitively associate with the Web site; for example, in the case of the New York Times, that root provider is www.nytimes.com.

For this analysis, we chose eight candidate root servers based on three categories: (i) *media and broadcasting* (e.g., media streaming associated with non-web media companies); (ii) *entertainment* (e.g., communal streaming sites); and (iii) *pornographic* (e.g., sexually explicit sites).

Each of these categories (media, entertainment, and pornography) represent a different risk profile for the DoD. Media and broadcasting sites may be considered conventionally safe—in that they provide published, forwarded information and are a known quantity. Media sites are the sites that the DoD arguably have the most interest in allowing to stream to the NIPRNet. Entertainment sites provide user-generated content, such as video blogging.

DoD may have interest in controlling or filtering this traffic, especially in combat zones or on sub-networks such as Afghan Mission Network (AMN). Finally, the DoD has a zero tolerance policy against pornographic sites.

We collected sample datasets with full packet traces by browsing a random subset of popular streaming sites in August 2010. There were two requirements used for comprising a dataset:

- Random selection of video-streaming Web sites.
- Predetermined benchmark testing periods.

We used Wireshark and tcpdump to capture the packets. In our collections, we were able to follow the default-filtering guidelines for these two applications in building the datasets.

1. Web site Selection

The candidate sites were chosen from the top video sites identified by Alexa (www.alexa.com), which provides worldwide Web site ranking. Ranking is not solely limited to the video streaming category, but includes Web sites providing other services as well. Web sites were further categorized by tags for "video sharing or hosting" to assist in multimedia identification. Content was divided into 3 categories for analysis:

- Media (e.g., news and sports broadcasting)
- Entertainment (e.g., movie and video clips)
- Pornographic

Web site	Alexa Ranking	Category
www.youtube.com	3	Entertainment
www.xvideos.com	56	Pornographic
www.pornhub.com	58	Pornographic
www.youporn.com	69	Pornographic
www.espn.go.com	95	Media
www.tube8.com	94	Pornographic
www.dailymotion.com	106	Entertainment
www.hulu.com	224	Entertainment
www.foxnews.com	1,297	Media
www.moviefreak.com	834,136	Entertainment

Table 2. Names, rankings and categories of test site provided by Alexa

2. Collection Process

To observe traffic patterns, data was collected over a 3-day period. The collection process involved navigating to each site and watching video clips for at least a 10 minutes. Streaming segments from Media sites were shorter than the desired duration and required multiple sessions to achieve our duration requirement. To minimize extra data, we initiated no other Web activity during the capture. The data collected was saved in a raw packet capture (pcap) file for further analysis.

ID	Root Provider	Date	Bytes	Packets	Unique IPs
DM1	http://www.dailymotion.com/	12AUG2010	25552241	25052	10
DM2	http://www.dailymotion.com/	13AUG2010	28450765	28749	8
DM3	http://www.dailymotion.com/	14AUG2010	61746825	58525	8
EN1	http:// espn.go.com	12AUG2010	28091642	29820	8
EN2	http:// espn.go.com	13AUG2010	15866515	15521	20
EN3	http:// espn.go.com	14AUG2010	22900658	22275	12
FN1	http://www.foxnews.com/	12AUG2010	27983071	30729	25
FN2	http://www.foxnews.com/	13AUG2010	51748597	57907	5
FN3	http://www.foxnews.com/	14AUG2010	59818789	65366	16
HU1	http://www.hulu.com/	12AUG2010	63420284	66470	7
HU2	http://www.hulu.com/	13AUG2010	57130957	60166	7
HU3	http://www.hulu.com/	14AUG2010	55276216	64187	1
MF1	http://www.moviefreaker.com/	12AUG2010	59553645	56122	8
MF2	http://www.moviefreaker.com/	13AUG2010	34491111	38517	7
MF3	http://www.moviefreaker.com/	14AUG2010	31129589	30007	30
PH1	http://www.pornhub.com/	12AUG2010	33127872	33911	9
PH2	http://www.pornhub.com/	13AUG2010	50009115	50428	6
PH3	http://www.pornhub.com/	14AUG2010	48431632	49409	8
TB1	http://www.tube8.com/	12AUG2010	20242362	19277	6
TB2	http://www.tube8.com/	13AUG2010	43398896	42010	28
TB3	http://www.tube8.com/	14AUG2010	23971477	21865	6
XV1	http://www.xvideos.com/	12AUG2010	55454579	60875	77
XV2	http://www.xvideos.com/	13AUG2010	59409325	61260	32
XV3	http://www.xvideos.com/	14AUG2010	39221867	40852	9
YP1	http://www.youporn.com/	12AUG2010	49816165	46726	12
YP2	http://www.youporn.com/	13AUG2010	28941351	27700	6
YT1	http://www.youtube.com/	12AUG2010	39202110	36003	34
YT2	http://www.youtube.com/	14AUG2010	76196863	70642	9
YT3	http://www.youtube.com/	13AUG2010	80617513	73327	8

Table 3. Initial data capture

B. ANALYSIS

1. Classification by IP Address

Our first method of analysis was to classify streaming Web sites by IP addresses. We started by querying the local DNS server using the command line *nslookup* tool and recorded the results for comparison with captured data. Wireshark was used to examine the pcap file to isolate which IP address was delivering streaming content. Wireshark contains an option to filter a capture file by conversation. A Wireshark conversation consists of the two-way traffic between two IP addresses. This allowed us to sort the conversations by IP address and compare addresses from Wireshark data to the nslookup results. With the exception of the DM3 sample associated with www.dailymotion.com, conservations from Wireshark do not contain addresses in the *root provider's*

registered IP address space, indicating data was streaming from another source. We connected to an external server and performed an *nslookup* to determine discrepancies. The results produced were different from the initial DNS query. Reverse lookups were performed using the Reverse Lookup tool available on the American Registry for Internet Numbers (AIRN) Web site. (ARIN 2010) The results confirmed both DNS requests produced valid addresses for the root provider, but due to querying separate DNS servers, we received IP addresses located in different networks.

We concluded from the session that the largest bytes being transferred were most likely streaming video. This session was further compared to expected duration of the viewed video stream. The results produced the IP address serving the video stream, as depicted in Table 4.

Server	Client → Server	Server→ Client	Duration (Sec)
74.125.224.27	3928	27776	592
74.125.93.121	3287	41983	332
74.125.164.145	1294524	73327052	560

Table 4. Determination of video content in data from YouTube 13Aug capture. We verified that the session with largest Sever to Client transfer was the streaming video

The list of IP addresses was compared to DNS lookup results, which again revealed the root provider's address was located in a different IP range than the server delivering the content. We performed the ARIN lookup with the new IP addresses, recorded the network range, and registered owner, as shown in Table 5. This testing strategy revealed that the video was streamed from third-party sites, indicating either Proxy or Content Distribution Networks (CDN) methods. Subsequent sections will discuss Proxy and CDN in detail.

Site	IP address	Registered Owner	NetRange
http://www.dailymotion.com	96.17.147.99	Akamai Technologies (AKAMAI)	96.16.0.0/15
	188.65.120.5	Dailymotion S.A.	188.65.120.0/21
http://espn.go.com	72.247.218.90	Akamai Technologies (AKAMAI)	72.246.0.0/15
	63.217.232.42	Beyond The Network America, Inc.	63.216.0.0/13
http://www.foxnews.com	64.212.60.145	Global Crossing	64.212.0.0/14
	216.156.211.40	XO Communications	216.156.0.0/16
	204.10.29.120	Akamai Technologies (AKAMAI)	204.10.28.0/22
http://www.hulu.com	65.49.92.212	Hurricane Electric, Inc.	65.49.0.0/17
	8.12.221.122	Level 3 Communications, Inc. (LVLT)	8.0.0.0/8
http://www.moviefreaker.com	209.222.128.137	Carpathia Hosting, Inc.	209.222.128.0/19
	74.125.166.153	Google Inc.	74.125.0.0/16
	173.194.12.94	Google Inc.	173.194.0.0/16
http://www.pornhub.com	204.93.184.70	Cognitive Networks Inc. (COGNI-6)	204.93.184.0/22
	216.18.184.206	Reflected Networks, Inc. (REFLE-2)	216.18.160.0/19
	216.28.184.202	Cogent Communications (COGC)	216.28.0.0/15
http://www.tube8.com	216.18.165.198	Reflected Networks, Inc. (REFLE-2)	216.18.160.0/19
http://www.xvideos.com	208.111.173.105	Limelight Networks, Inc. (LLNW)	208.111.128.0/18
http://www.youporn.com	205.128.85.126	Level 3 Communications, Inc. (LVLT)	205.128.0.0/14
http://www.youtube.com	74.125.164.145	Google Inc.	74.125.0.0/16

Table 5. Video Content Servers

a. Proxy Server

Proxy-assisted delivery systems have emerged and are expanding rapidly. Proxy servers exploit their processing and buffering capabilities to provide network-wide video streaming in a coordinated and distributed manner. At the core of the proxy system resides one or more caching servers with a large data repository as depicted in Figure 6. Proxy servers are placed across the client's network, typically attached to gateway routers connecting a campus network to the wide-area network (WAN) and the Internet. The proxy servers reduce upstream bandwidth by caching frequently accessed data. This allows clients within the campus network to retrieve the data from the proxy server vice the root provider.

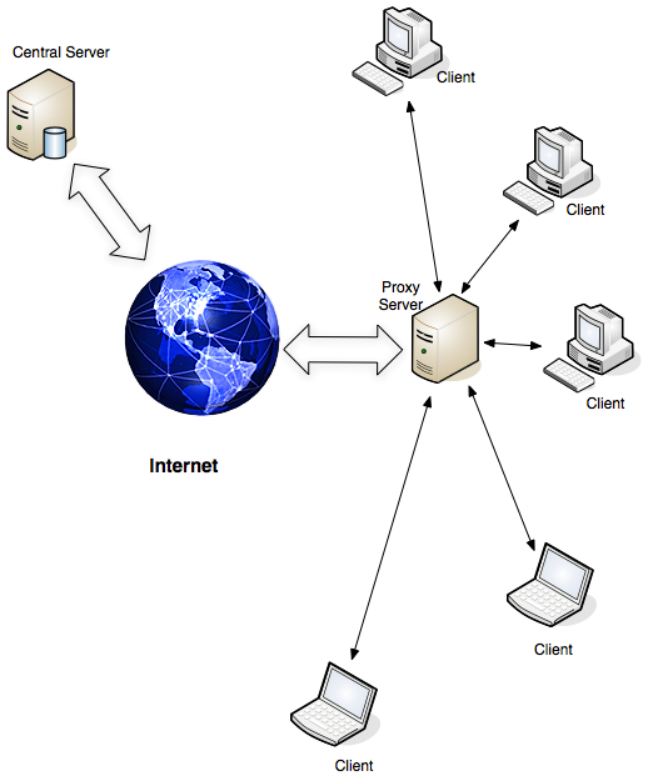

Figure 6. Typical Proxy Server

b. Content Distribution Networks

Some Web sites observed use Content Distribution Networks (CDNs) as a means to deliver large amounts of data. This data is placed throughout various points of the Internet in order to minimize access latency. This allows the client to access data from the closest CDN as opposed to accessing the root provider.

CDNs operate on the principle of locating the mirror server closest to the user in a network sense, if not geographically. This strategy alleviates potential bandwidth or processing bottlenecks at or near the root provider server. Content types may include web objects, downloadable objects (e.g., media files, software, and media streams), as well as other components of Internet delivery (DNS, routes, and database queries). CDNs are typically owned by a third party (e.g., Akamai Inc.) and differ from proxy servers by including a complete

replication of data across their servers. A caching or proxy server is provided by the client's network and normally contains only the most popular data accessed from their network as opposed to a complete mirror.

As an example, several Web sites analyzed during this study used Akamai and Limelight as CDNs. Sites rename their URLs with a specific prefix. Resolving the hostname strings using the Domain Name Service (DNS) yielded an IP address of the CDN server. CDN mirroring must ensure that the DNS lookup tries to yield the nearest mirror site. CDNs are largely proprietary, but a general model is depicted in Figure 7.

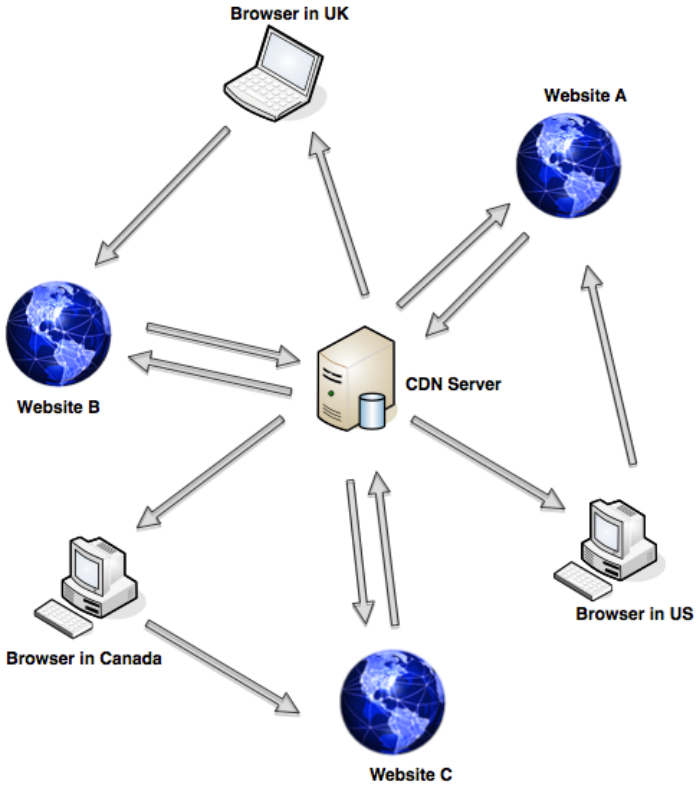

Figure 7. Typical video streaming under the CDN model

c. Limitations of IP Based Classification

We initially hypothesized that classifying video streaming sites by their IP address would assist in developing a signature to use in production testing. Several obstacles prevented this further investigation. First, upstream distributers of Internet numbers allotted multiple IP addresses to our target Web sites. Second, attempting to filter or classify based solely on an IP address would prompt targeted Web sites to either change ISPs or purchase additional IP addresses. This will result in an increase of man-hours to identify new addresses and add them to the signature file for blocking or limiting access. Another shortcoming with blocking a CDN streaming video is that the relationship between a CDN and a streaming provider is *not* 1:1, and as a result, blocking a particular CDN means blocking all content provided by that particular CDN, *even if* DoD wants that authorizes that specific content.

The next issue was unexpected and serves as an area for further research. We expected to identify multiple streaming protocols in use and apply their signatures in distinguishing streaming video from normal traffic. Streaming Protocols such as RTP, RTCP, and P2P all use User Datagram Protocol (UDP) to transmit media content. UDP is a connection-less protocol and thus does not overload the server with processing acknowledgement packets from clients' currently streaming video. The disadvantage is UDP does not process lost or corrupted packets. For live video streaming, this is normally not an issue. The feed may fault, but will return to normal quickly, providing lost packets were not indicative of larger networking issues. UDP is also easy for system administers to filter, since the majority of Internet traffic uses TCP as the primary delivery protocol. The only significant UDP traffic in our testing datasets was DNS queries (Port 53). This is a standard function and is required for computers to translate a domain name into an IP address.

We did observe two instances of a TCP streaming protocol in use. The August 13 and 14 captures from Hulu show a TCP source port of 1935.

According to The Internet Assigned Numbers Authority (IANA), port 1935 is reserved for macromedia-fcs, described as Macromedia Flash Communications Server MX. (Kohler, Handley and Floyd 2010) Adobe, the owner of Macromedia Flash confirms their server platforms use RTMP on port 1935 to distribute streaming content. (Adobe n.d.) Adobe's documentation also states other ports may be used to circumvent firewalls. We recommend that system administrators filter port 1935 and only allow approved sites to connect on that port. This thesis will only focus on HTTP streaming, which uses TCP.

Several of the sites profiled, utilized more than one CDN. For example, Fox News uses Global Crossing, XO Communications, and Akamai Technologies to distribute their content. Our testing prompted the association between streaming Web sites and individual CDNs. Due to privacy agreements, we were unable to ascertain a comprehensive list of CDN customers. We performed a Google search and noted several third party sites advertising to sell a CDNs membership listing, but we questioned the reliability of the information due to it not coming from the CDN owner. Another concept suggested certain types of web sites favored a particular CDN. We theorized that some companies might not enter into a contract with a CDN distributing pornography. Our motivation was to block CDNs that only provided pornography, thus eliminating a source of streaming and prohibited content on the DoD networks. Our sample size was not large enough to make this determination. We found two pornography sites, as well as Hulu using the same CDN. Hulu is primary a video streaming site, but their content is not pornography. The DoD is faced with the problem of either blocking all CDNs that serve pornography, which would also deny access to legitimate content (e.g., News reports) that uses the same CDN, or block CDNs that do not currently host approved DoD content. The issue with blocking CDNs is that the CDN does not own the content. The content provider can either switch or add additional CDNs to bypass the filter. With this approach,

the DoD will eventually block some amount of desired content. This led us to conclude that without a reliable list of a CDN's customers, blocking a CDN may affect legitimate military browsing.

2. Classification by Keywords

Words frequently appearing in capture files are termed Keywords and are the next element examined in the full packet captures. During initial examination, we noticed certain keywords were present in the full captures. We parsed the *pcap* files using Wireshark and recorded frequent occurring terms that may be indicative of streaming video. The following terms were chosen based on observation of the datasets: "HTTP: Continuation or non-HTTP traffic", "akamai", "video", "player", "porn", "swf", and "flv."

The "HTTP: Continuation or non-HTTP traffic" message is indicative of a packet received on Port 80 that does not contain a HTTP header. IANA classifies port 80 as a well-known port and reserved for HTTP traffic. This is a recommended standard, but nothing prevents other applications from sending and receiving data on port 80 (e.g., HTTP encapsulation or tunneling). The message also indicates that multiple HTTP packets were required to deliver the content. The Maximum Transmission Unit, or MTU is the largest link-layer frame (in bytes) that can be sent from a given network device. Ethernet Version 2, defined by the IEEE 802.3 standards, defines a MTU of 1518 bytes per frame including the link-layer header. A standard TCP/IP packet transmitted over Ethernet contains a 14 byte Ethernet header, a 4 byte Ethernet trailer, 20 byte IP header, 20 byte TCP header, and 1460 bytes for payload. If more than 1460 bytes of TCP payload are required to deliver the requested content, the server must send more than one packet. It is up to the server whether to retransmit the HTTP header or to rely on the reliable data transfer properties of TCP to reconstruct the data for the client's application.

We took three samples of HTTP traffic from sites without streaming media. In the three aforementioned tests, the above message does not appear. The terms: video, player, porn, swf, and flv were found in ten of the HTTP *Get* Requests. The terms *video* and *porn* were chosen to find content using those terms in the request path. The *player* term occurred when the server sent streaming parameters to the player on the client's computer. We chose Swf and flv, two popular formats for HTTP streaming video. The following example of a HTTP Get Request was found on the EN1 capture:

"GET/p/espn_live/VideoAdRenderer.swfHTTP/1.1\r\n."

The term "akamai" was found frequently in DNS requests, showing the client was querying the DNS server for an IP address of "akamai" and additional content was being requested from the CDN.

The *frame* field was used to classify a session by packet size. We observed that the majority of streaming video packets contained the maximum TCP data segment equal to 1460 bytes, and counted the number of packets per session that matched this value.

In order to parse all the captured files from our datasets quickly and efficiently, a Perl script "keywords.pl" was created (See Appendix for source code) that would open each file, search and count terms, collect IP addresses, and record the information. We discovered that pcap files are in a binary format, which does not allow direct parsing. To get a text format, we used Wireshark to produce flow graphs. A flow graph depicts traffic flow during the capture period, as represented in Table 6.

Our program produced two files for further analysis. The first file consisted of a table displaying counts for a previously identified term, while the second contained every publicly routable IP address found in the flow graph. Table 7 lists a sample of the output.

Time	208.111.148.6<----------->172.16.1.10	
0.000	Continuation or non	HTTP: Continuation or non-HTTP traffic
	(80) ----------> (49681)	
0.000	49681 > http [ACK]	TCP: 49681 > http [ACK] Seq=1 Ack=1441 Win=32400 Len=0 TSV=192631709 TSER=1884648797
	(80) <---------- (49681)	
0.012	Continuation or non	HTTP: Continuation or non-HTTP traffic
	(80) ----------> (49681)	
0.012	49681 > http [ACK]	TCP: 49681 > http [ACK] Seq=1 Ack=2881 Win=33120 Len=0 TSV=192631709 TSER=1884648797
	(80) <---------- (49681)	
0.024	Continuation or non	HTTP: Continuation or non-HTTP traffic
	(80) ----------> (49681)	
0.024	49681 > http [ACK]	TCP: 49681 > http [ACK] Seq=1 Ack=4321 Win=32400 Len=0 TSV=192631709 TSER=1884648818

Table 6. Sections of a flow graph from EN3 dataset.

The *keyword* search and maximum frame count did not produce solid results. Table 7 depicts the variance across the entire sample with no combination of categories clearly indicating streaming video. Samples taken from the same root provider produced inconsistent results as well. The FN1 and the HU3 samples contained streaming video, but their counts were similar to the normal surfing control samples.

Using Wireshark, we searched all DNS queries and *HTTP GET* Requests and discovered the majority of identified keywords were contained in advertisement requests. Up to 75 unique IP addresses were found per captured session with average counts of 14.55 addresses. The majority of these sources were serving content unrelated to a streaming video session. The following is a *HTTP GET* Request from a FN3 sample:

"GET /promos/061410_Yoplait_Kitchen_FNC_LOW.mp4 HTTP/1.1\r\n.

This is an advertisement for Yoplait yogurt that played during the streaming session. We did not request this video; however, it was automatically

downloaded to the browser. We adjusted our script to filter lines containing “ad” and conducted another test, resulting in the similar counts.

	Search Terms								
Dataset	**HTTP**	**akamai**	**video**	**player**	**porn**	**swf**	**flv**	**frames**	**Unique IPs**
DM1	564	2	3	0	0	1	0	242	10
EN1	568	4	1	0	0	1	0	351	8
EN2	11	8	10	0	0	3	0	236	20
FN1	0	6	2	0	0	2	0	305	24
FN2	604	0	0	0	0	0	0	371	5
HU1	582	5	0	1	0	0	0	322	7
HU3	0	0	0	0	0	0	0	360	4
No Streaming 1	0	13	0	0	0	0	0	284	*
No Streaming 2	0	4	0	0	0	0	0	352	*
PH2	613	4	0	0	0	0	0	278	6
PH3	67	0	0	0	0	0	0	353	8
TB1	671	0	0	0	0	0	0	300	6
TB2	0	8	3	2	0	3	0	196	28
XV2	0	7	42	0	6	0	1	236	75
XV3	0	0	5	0	2	0	1	271	9
YP1	386	5	1	0	1	1	0	235	12
YP2	695	0	0	0	0	0	0	305	7
YT1	575	0	10	1	0	0	0	229	34

Table 7. Keyword and Unique IP counts generated by “keywords.pl”
***** unique IPs not counted due to the absence of streaming video

The final analysis of our *keyword* search was to use packet captures to determine which video was viewed during this session. We found that by examining an advertisement request packet, we could read the *HTTP Referrer* field and extract the link to the associated video. This information, along with a cookie is forwarded to an advertisement server to identify the client, as well as select an advertisement message. In a *Referrer* field from the DM3 sample, we observed the following:

http://www.dailymotion.com/video/x8pk76_the-outlaw-emmett-deemus-the-porno_shortfilms.

This allows us to observe which video was requested, as well as a link path for viewing the video. Only a few of the sites in our samples place the full video path in the *Referrer* field. In the case of the XV3 sample, the *Referrer* field is populated with the following contents:

http://www.xvideoslive.com/?DF=0&AFNO=1-0-611418338312&UHNSMTY=438.

This information allows navigating the link to view the streaming video. Apart from the domain name, it gives no clear indication of what type content was requested from the server. Web sites also use unique session cookies in combination with requests to ensure links can only be used during that session. This prevents another site from posting links directly to a server's content, leading to the server missing advertisement revenue when its content is streamed from a third party.

a. Limitations

Further testing revealed some limitations with keyword searches. Encoded filenames may not be streaming content, but advertisements that skew the results, requiring a deep packet inspection to get additional information. In perspective, we collected approximately 1.35 GB of data over three days to form our datasets, roughly 100 minutes of video streaming per day by a single user. Based on our Internet browsing habits and use of streaming video, Voice over IP (VoIP), and other online activities, we consumed more than our entire data collection in one evening. The DoD has approximately 3.1 million civilian and military personnel (Military 2010). If each user consumed the daily amount of our test set, DISA would need roughly 1.395 PB (1.305×10^{15} bytes) for just one day of full content collection. This is costly in terms of required resources for data warehouses and produces more data than can be analyzed.

Encoded filenames prevent discriminating between advertisements, streaming video, or traditional static Web site data. Testing allowed identification of DNS queries in the traffic streams; however, this alone does not provide the

system administrator with information to block known CDNs and streaming sites. This method is already being evaluated throughout DoD and will not be explored further in this thesis. We also feel that with the increase of Digital Rights Management (DRM), privacy concerns, and encrypted traffic, keyword profiling provides limited and inconclusive results.

3. Classification by IO Plots

Building on what we learned in previous two sections of this chapter, we moved towards analyzing statistical data from each capture in an attempt to find signatures that identify streaming video. In this section, analysis will continue with testing data sets minus the deep packet inspection. Wireshark is used to produce statistical data from each session, as well as Input/Output (I/O) graphs.

The first question we had when examining statistical data from known stream video sessions was, “How to differentiate streaming video from a normal download?” Many DoD approved sites allow the download of large files. For example, Navy Knowledge Online (NKO) offers audio books and other types of downloadable media. Using the data from previous analysis revealed limitations of IP or keyword classification, thus prompting our decision to investigate signature differences between downloaded traffic and streaming video.

Our first attempt was to use Wireshark to do a graphical representation of flow data. Wireshark has the ability to produce an IO graph from pcap files. This graph plots the bits transferred per second (bps) and gives a visual depiction of data transmission. We augmented the testing dataset with a capture that does not contain streaming video and a capture of a 666 MB “Pentoo Linux LiveCD.” A Linux LiveCD is a complete operating system and associated applications that can run completely from CD/DVD without hard drive access. We chose this file due to its size and the ability to download it over HTTP. One alternative we explored was to generate and capture large amounts of random data. We decided against further research in this area, since it would not involve network

congestion, buffering, or other networking aspects captured in the other samples. At first glance, it was easy to distinguish three different types of content. The normal browsing sample in Figure 8, illustrates a HTTP Get Request, the server's response, followed by the users digesting delivered content. The user then requests a different link and the process repeats. Figure 9 represents a streaming video feed from TB1. In this plot, we observe the TCP flow control mechanism and buffer of the client's streaming video player. As data is received, the player's buffer fills in size until the data rate reaches a peak. Once the buffer is filled, the client's player returns a window size of zero, thus notifying the server to stopped sending data. The server sets an internal timer to wait before trying to resume data transfer. When this timer expires, the server sends a test packet to determine if the client is ready to receive another packet. This is seen by the peak and down slopes in Figure 9. Once the client is ready to receive, the process resumes. In Figure 10, we request a Linux disk image from

http://mirror.switch.ch/ftp/mirror/pentoo/pentoo-i686-2009.0.iso.

The file size is 666MB and constitutes the largest capture performed for this thesis. The same TCP flow controls are seen, but with a faster recovery due to the lack of a streaming video buffer. The data rate falloff stops as soon as the internal network buffer of the operating system is able to resume processing incoming packets.

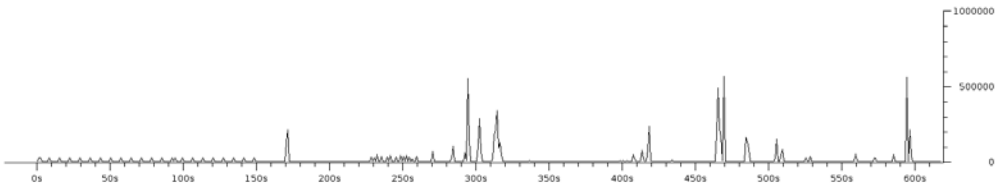

Figure 8. Non-streaming browsing where the X-axis is the session duration in seconds and the Y-axis is bits per second. Note the peaks followed by no traffic.

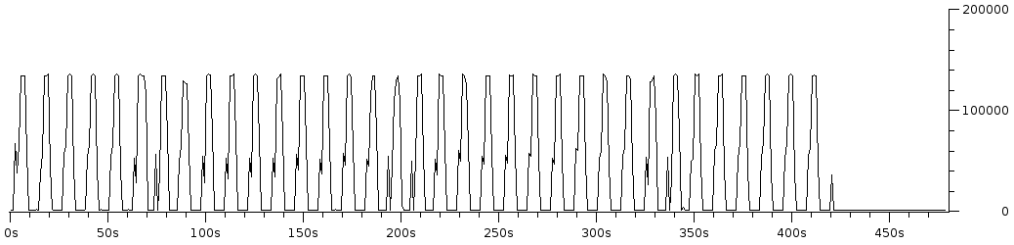

Figure 9. Tube8 12 Aug sample where the X-axis is the session duration in seconds and the Y-axis is bits per second. In this figure, the pattern repeats at a constant rate until the stream finishes.

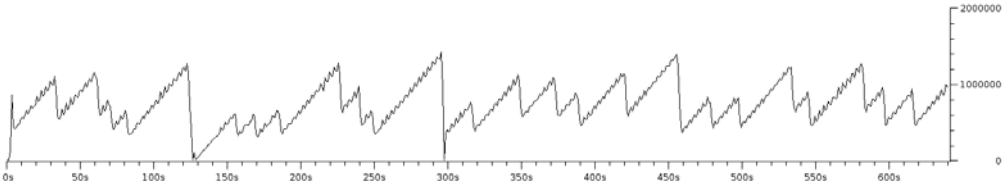

Figure 10. 666MB Linux ISO download where the X-axis is the session duration in seconds and the Y-axis is bits per second. This shows the filling of the systems buffer, the TCP back-of, and the resumption of the file transfer.

With this information, we were able to predict normal browsing vs. streaming. We ran into problems, however, trying to separate HTTP streaming from a normal file transfer. During our testing, we observed samples from our datasets occurring between the patterns shown in Figures 9 and 10. This led to the conclusion that we could not use this method of visual analysis to predict content with a reasonable level of accuracy. The other obstacle with using IO plots to predict content is that data from each packet is required. The plots in Figures 8-10 are generated by examining the arrival time and TCP data segment size of each packet received. This means, each router or a designated flow capture device would have to track every single packet from all the incoming sessions to generate the above figures. This would require a large processing resource to produce a plot for each flow. This could be automated, but the collection and overhead would limit its usefulness.

4. Asymmetric Transfer Characteristics

During our research for section three, we reviewed a published technical paper titled “Fine-grained traffic classification with Netflow data.” (Rossi and Silvio Valenti) Research centered on classifying applications used for Peer-to-Peer (P2P) streaming video. We contacted the authors and received permission to experiment with a demo version of their Abacus algorithm. (D. Rossi n.d.) The Abacus algorithm analyzes relationships between control packets used to set up P2P networks and the following data packet sizes. Each application examined used a different size of control and data packets, as well as a unique ratio between the two types. This signature enabled the Abacus algorithm to predict the particular application. We ran their algorithm against their provided sample and evaluated the results. It was highly successful in predicting P2P streaming, but failed to detect streaming flows in our samples. Closer analysis of the provided test sample revealed their research focused on UDP traffic. Many P2P applications utilize UDP to reduce overhead incurred when a TCP connection is established and acknowledgments sent. P2P efficiency relies on small data transfers from many simultaneous sources. The TCP overhead would reduce the efficiency of P2P, as well as the benefits of streaming over P2P.

Following Rossi and Valenti’s analysis of Netflow data, we decided to examine the same parameters on our samples to determine a correlation to streaming video. Wireshark was used to produce Table 8, consisting of one dataset and the two-way traffic conducted during the streaming session. From Table 8, we noted the disparity between download/upload traffic. Each download was greater than 10 MB, while all uploads were significantly less. The client requests information using an HTTP Get Request and the server responds with the information and, perhaps, a few advertisements. The field that sets this data apart from normal browsing behavior is the duration of the session. In each of these sessions, the disproportion of data occurs in approximately 10 minutes. This indicated that the server was continuing to provide large amounts of data to

the client per session, relative to non-streaming content. With this metric, we are able to determine a non-streaming session from HTTP streaming /file transfer.

Dataset	Download (KB)	Upload (KB)	Duration (sec)
DM1	24558	478	188.96
EN1	24029	651	219.11
FN1	21671	874	311.79
HU1	60549	1762	708.23
MF1	13409	264	227.24
PH1	31854	704	403.80
TB1	19267	368	437.26
XV1	49590	1322	558.80
YP1	47833	832	369.75
YT1	37762	631	368.20

Table 8. Upstream and Downstream bytes and duration for selected datasets. Note the high download:upload ratio

Our next challenge was to differentiate between HTTP streaming and file transfer. As seen is Figures 9 and 10, both utilized TCP and have similar characteristics due to TCP flow control. To assist in discriminating usage, we captured downloads of three Linux disk images with varying sizes from three separate servers as listed in Table 9. The first characteristic we noticed was the relationship of bytes transferred to the duration of the session. With the exception of the Rapidshare session, the file downloads were transferring data approximately five times that of the fastest streaming session. Rapidshare is an Internet based hosting and distribution network. Users upload files to a password-protected account and allow other users to download those same files. We examined the Rapidshare data to determine why it did not follow the same pattern and determined that downloads were throttled for their non-paying customers. To maximize bandwidth, a membership fee is required to download at full speed.

	Download (bytes)	Upload (bytes)	Duration (seconds)
666MB from Pentoo	731904482	17266460	1016
151MB from Gentoo	163251741	3670800	217
96MB from Rapidshare	105197394	2776566	1278

Table 9. Upstream and Downstream for File transfers

a. Limitations of Asymmetric Transfer Characteristics

The major limitation discovered is the ambiguity between streaming video and a file transfer. The file transferred may be media, a software patch, or approved content from sites such as NKO. Without examining the IP address or conducting a deep packet inspection, it is difficult to determine which type of content is present. For system administrators, a file download may have less of an impact on the network. This is due to the shorter duration per file because of the increased bit rate compared to streaming.

We attempted to obtain the minimum and maximum streaming bit rate for the Adobe Flash server to determine where to set our own signatures for classification. Our research of the Adobe Flash Server initialization and deployment documentation led to the conclusion that these settings are server dependent and implemented by the system administrators during configuration. Bandwidth availability and demand are the two primary considerations in setting the streaming bandwidth. Without an average value from the manufacturer, we determine in the next section the statistical separation of streaming content from file downloads from our datasets.

5. SiLK

Our next step was to determine the ratio of downloaded bytes to session duration that would indicate streaming content. We decided to convert our datasets to the SiLK format for additional analysis. We used Yet Another Flow-Generator (YAF) from Carnegie Mellon University's Computer Emergency Response Team (CERT) NetSA Security Suite to convert all of our pcap sessions into SiLK format. We then employed the SiLK command *rwflowpack* to

read our sample data and convert it to SiLK flow data. The command *rwfilter* was used to filter incoming data. This result was passed to *rwstats* to generate statistics, as seen is Table 10.

sIP	sPort	dIP	dPort	packets	bytes	dur
130.59.10.36	80	10.10.10.4	58813	511293	724741850	1015
156.56.247.195	80	10.10.10.4	56685	107827	161740500	217
62.67.0.28	80	172.20.194.57	39822	73714	104164793	1277
96.17.69.143	1935	10.10.10.4	35082	68518	80192825	687
74.125.164.155	80	172.16.1.10	52359	51920	76729648	601
74.125.164.145	80	172.16.1.10	54089	48702	72645121	560
208.111.148.111	80	172.16.1.10	51360	37831	55878751	607

Table 10. Rwstats output from SiLK

We sorted our SiLK dataset using bps and observed the known streaming content grouped together between 1.2 Mbps and 0.3 Mbps, as depicted in Figure 11. There remained several samples scattered in between our known streaming data. We concluded that these were the advertisements viewed earlier by examining their duration. All samples had durations less than 45 seconds and a median session size of 3 MB. When we filtered datasets with these criteria, only two non-streaming samples remained in the group. The first from the file sharing site RapidShare, and the second was an encapsulated file transfer in a Secure Shell (SSH) session. RapidShare offers a free and paid tier of service. The free version restricted bandwidth to an average rate of 652,559 bps over the duration of 1,277 seconds. This falls squarely inside our testing criteria and cannot be filtered at this time, whereas the SSH session is filtered by the source port. Additionally, we set an upper asymptotic bound to 200 MB to correspond to upper limits of our testing datasets, not including the 666M ISO. The fields remaining are IP address, port, and the number of session packets. We discussed classification by IP addresses earlier and did not find applicable use for the packet field that was not already covered with the byte field.

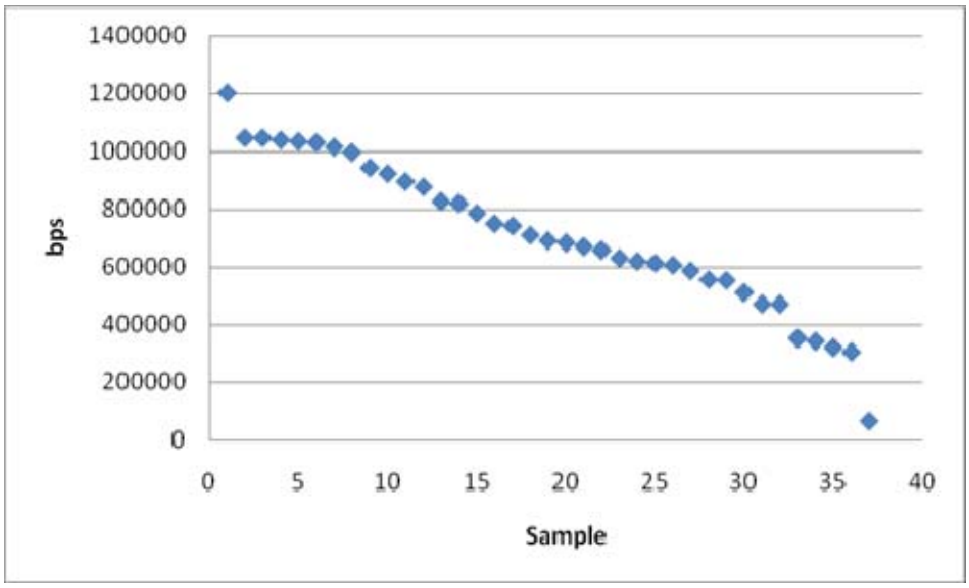

Table 11. Bps per sample. Note the grouping between 0.3Mbps and 1.2Mbps

C. SUMMARY

In this chapter, we investigated several different methods to detect and classify streaming media. Each method was explored and both advantages and disadvantages were discussed at length. We also transitioned from deep packet inspection to flow information in which only session statistics are available. Based on exhaustive testing, we determined the following criteria would offer the highest probability of detecting streaming video in the DISA Centaur datasets:

- Bits per second: > .3 Mbps and <1.2 Mbps
- Duration: > 45 seconds
- Session size: > 3 MB and < 200 MB
- Port: 80 (http) and 443 (https)_

These parameters form the basis of our detection algorithm. In Chapter V, we will incorporate them into a detection script and use it to predict streaming video from flow information collected from DISA repository databases.

V. CENTAUR ANALYSIS

This chapter presents validation of our algorithm using Centaur repositiory data and SiLK tools. SiLK uses the SSH protocol to connect to various analysis clusters, issue command-line search criteria, and provide statistics for further analysis.

We used SiLK to analyze 32 hours of NIPRNet flow records representing 825 GB of traffic. A total of 210,566 flow records were captured in two-hour intervals during this observation period. Subsequent sections will focus on extracting data obtained in flow analysis and apply heuristics and networking concepts presented in earlier sections to differentiate behavioral patterns.

A. FLOWS

A flow is a record of the one-way conversation between two networking devices: a client inside the flow generator's network and a server on the outside. Thus, a session will involve two flows: one in each direction. When the client initiates a session with a server, a flow is created to collect conversation statistics. When the server responds to the client's request, a second flow is created. Figure 11 represents an example of the flows generated during a TCP 3-way handshake.

- Flow 1 is created when the sensor observes the first packet between hosts A and B.
- Flow 2 is created when the server responds to the client's request.
- When the third packet is sent from the client to the server, Flow 1 is updated.

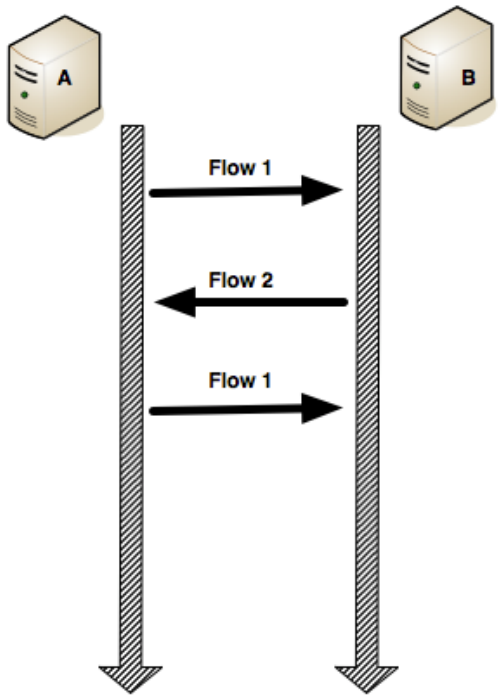

Figure 11. Flows generated. Flow 1 is from A→B and Flow 2 is from B→A.

Flows in the NIPRNet can be divided into three categories: inbound flows, which originate from an external host, outbound flows, and internal flows that do not involve any external host. Streaming traffic from proxy servers are internal flows under this classification. Given the relatively costly access link bandwidth, our analysis focuses on inbound flows and classifies the streaming traffic originating primarily from CDNs.

B. REPOSITORY QUERY

The starting point for querying Centaur repository databases is the *rwfilter* command. *Rwfilter* queries the flow repositories to match flows by input criteria and return the data to the user. *Rwfilter* requires three basic parts to perform a valid query, as seen in Figure 12.

- Selection Criteria
  - Repository: Time, Date, Class, Type, and Sensor

- Alternative: file or stream
- Partition
  - Filter Options: Selection of flows
- Output criteria

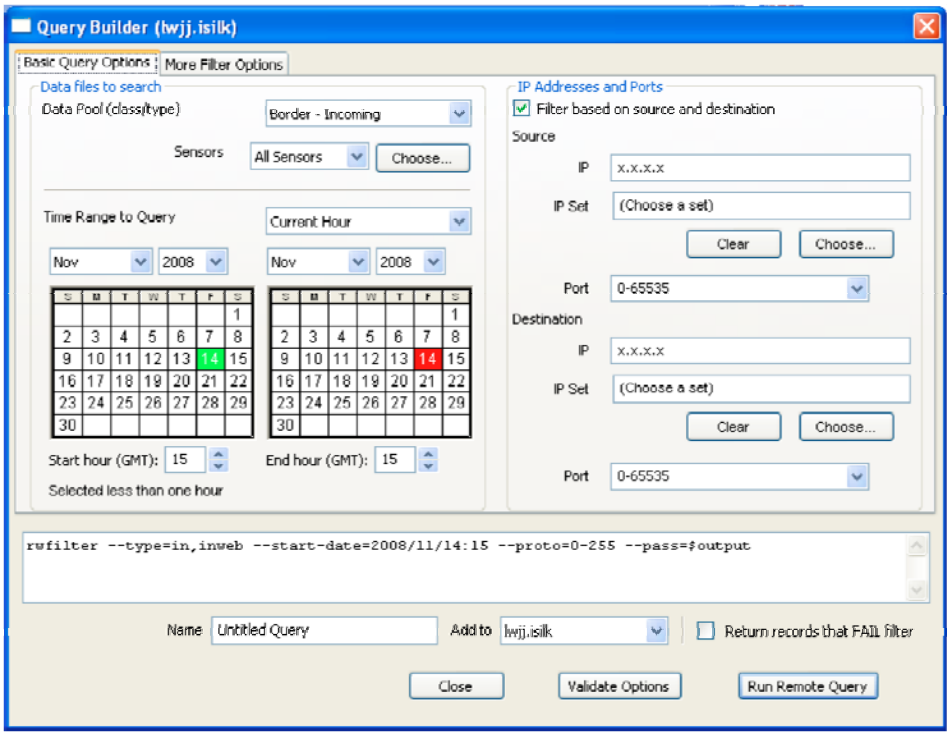

Figure 12. Query Builder for SiLK

Information is displayed as a problem set consisting of a collection of related flow data, analysis results, and visual graphs. Subsequent sections will provide amplifying details regarding analysis of the sampled flow sessions.

C. VISUAL ANALYSIS

We initially attempted to download 2.7 million flow records from Centaur, which corresponded to 48 hours of network traffic. Even with size and duration filters, the raw file size exceeded 600 MB and we were unable to download data

to our remote workstation. To affect sample downloads, we limited the maximum session size to 200 MB or less and requested 32 hours of flows, which produced 210, 566 distinct flow records.

The focus in this section was HTTP (Port 80), as well as HTTP Secure (Port 443). We used the Mathematica application suite to generate visual representations as depicted in Figures 13 and 14. To distinguish abnormal behavior from normal HTTP traffic, we evaluated 8 hours of Centaur flow data from August 12, 2010, which will be referred to as A12. We plotted bps versus flow identification with the 39 red dots indicating the testing samples used previously. The blue dots depict flows that met capture criteria: greater than 45 seconds in length and within the 3 to 200 MB asymptotic bounds. Cyan depicts flows less than 45-seconds in duration. Most streaming advertisements observed in our testing samples had a median duration of 38 seconds. Represented in Figure 14, emphasis was placed on three different minimum session duration times for comparison: orange = 20 seconds; yellow = 45 seconds; and purple = 80 seconds. This information reaffirms the decision to set a minimum duration of 45 seconds, excluding outliers such as advertisements and smaller streaming videos. As depicted in the Figure 14, the majority of our traffic sessions exceed 45 seconds in duration.

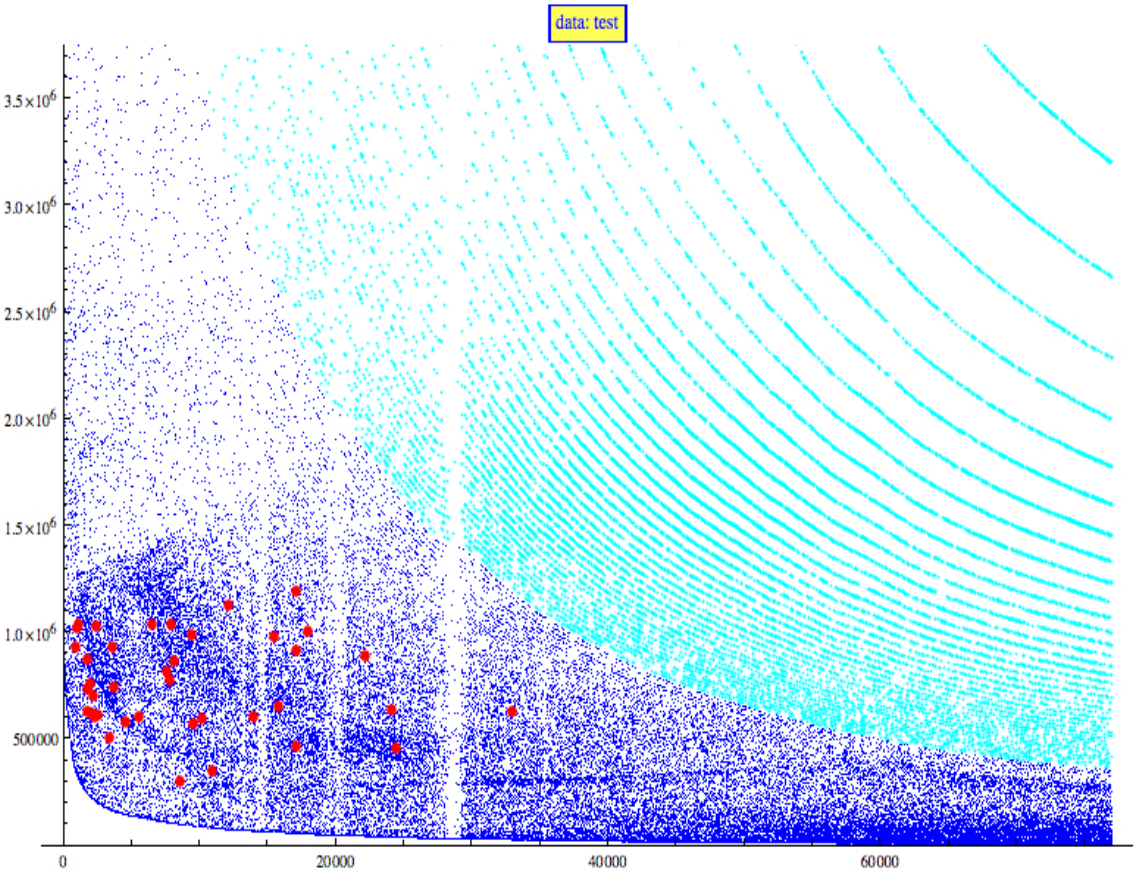

Figure 13. Flow bps of both testing data (red) and captured DISA data (blue >45 second duration and cyan < 45 seconds). The DISA data samples are sorted in decreasing session size while the testing data samples are placed at random positions between 0 and 35000 for visual effect.

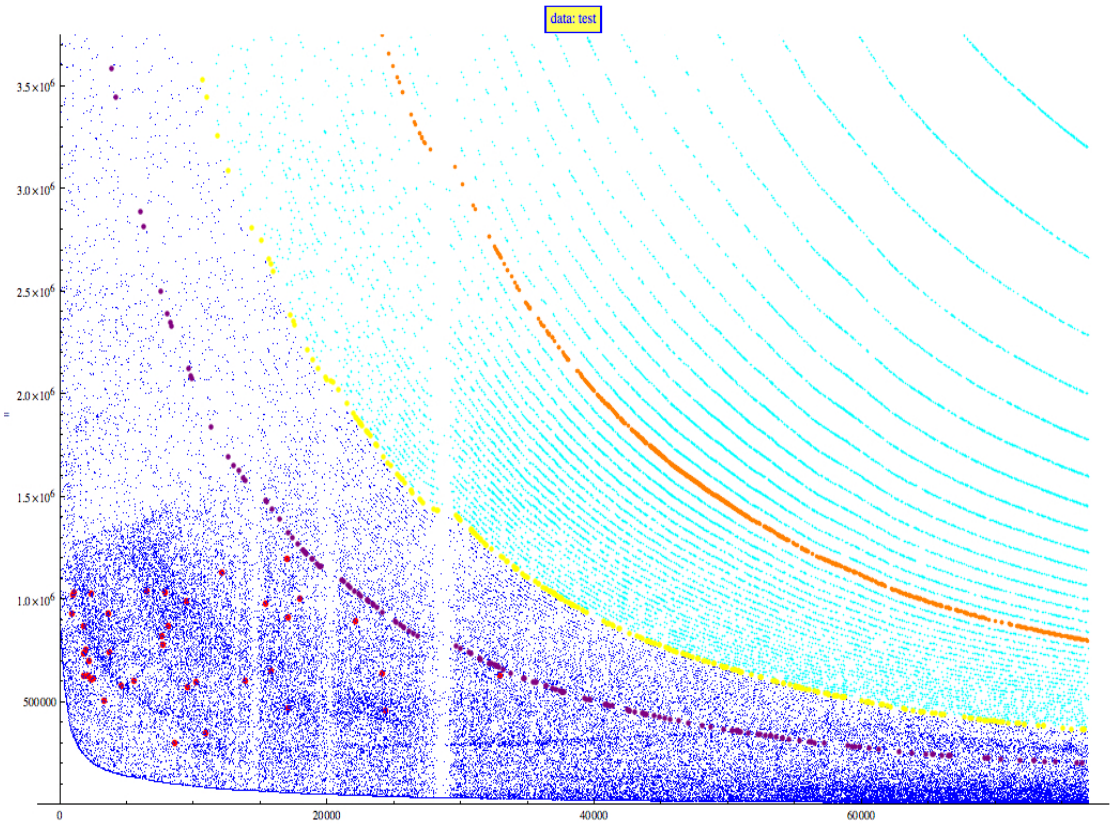

Figure 14. Flow bps vs sample ID of both testing data (red) and captured DISA data (blue and cyan). The orange curve represents a 20 second minimum duration, the yellow is 45 seconds and the purple 80 seconds.

D. VERIFICATION

The first step to verify our algorithm against collected flows from Centaur was to develop a script that parsed the entire sample, as well as manipulate the query parameters. The script "silk_Parser.pl" (Appendix), takes the statistics file produced above, examines each record, and filters based on criteria developed earlier in this thesis. The script outputs two files containing flows matching the filter, unique IP addresses, and statistics. The filter matched 21,425 out of 76,911 records, resulting in 3,240 unique IPs, from the A12 dataset.

To confirm the flow record most likely contained streaming content, we performed reverse DNS queries on the list of unique IPs. The goal was to

determine how many of the collected addresses matched our list of CDNs collected during testing. We wrote a Perl script “reverseDNS.pl” (Appendix), which reads a list of IP addresses from a file and requests a reverse DNS query from the local DNS server. The results are displayed in tabular form for further analysis. Not all IP addresses resolved to domain names. Some IP addresses were assigned to registries outside the contiguous United States (OCONUS) and did not resolve with our query. We reviewed the list of unresolved addresses in search of large blocks of contiguous IP addresses failing to return a domain name. For these IP address blocks, a manual query was performed using reverse DNS on the Arin Web site. This resulted in the name of the overseas registry that held information on that given IP range. One of the OCONUS registry sites was consulted and the results recorded. Arin was also used to identify all registered IP blocks owned by known CDNs discovered in this thesis.

Our script “silk_Parser.pl” was modified to compare the constructed list of CDNs to records matching our filter, as well as the addition of five counters for record purposes:

- Number of records in input file
- Number of records matching streaming criteria
- Number of records matching our CDN list
- Number of unique IP addresses in the matching records
- Number of unique IP addresses matching our CDN list

These counts were output to the display following execution of our “silk_Parser.pl” script, which predicted that less than 50% of A12 sessions were from a CDN. To refine our numbers, we took the unique IP addresses extracted from the A12 dataset that were unresolved and performed additional reverse DNS lookups. We repeated this process three times to create a list of 533 networks associated with video streaming, as well as inclusion of ISPs that had high concentrations of streaming originating from their networks. Where feasible, we attempted to resolve networks to the longest IP prefix of the content server to

minimize false-positive results by including networks not associated with streaming video. After final testing, we analyzed the impact of varying minimum session byte size. We used the same duration, bps, and maximum session size while varying the minimum session size from 2 MB to 20 MB. Results are shown in Figures 15 and 16. Figure 17 shows the relationship between varying the minimum session byte size and the percentage of CDNs predicted in the sample. At 10 MB, the matching records decreased because those not meeting minimum size were removed from the A12 sample. All other measurements decreased due to this modification. The percentage indicates records removed failed to meet our CDN classification, hence the overall ratio of CDNs to total sessions increased. This denotes either our minimum was set too low or a failure to identify several streaming media sites during our reverse lookups.

Flow samples indicate several blocks of IP addresses corresponding to ISPs offering Internet service. These ISPs also offer connections and content hosting to businesses. During DNS queries, we could not determine if some of the contiguous IP blocks hosted streaming content. Our classification decision was based on prior observation of streaming video originating from a network, as well as the number of sessions observed during this period that matched our criteria. Without additional streaming evidence, we choose to exclude these ISPs from our CDN list.

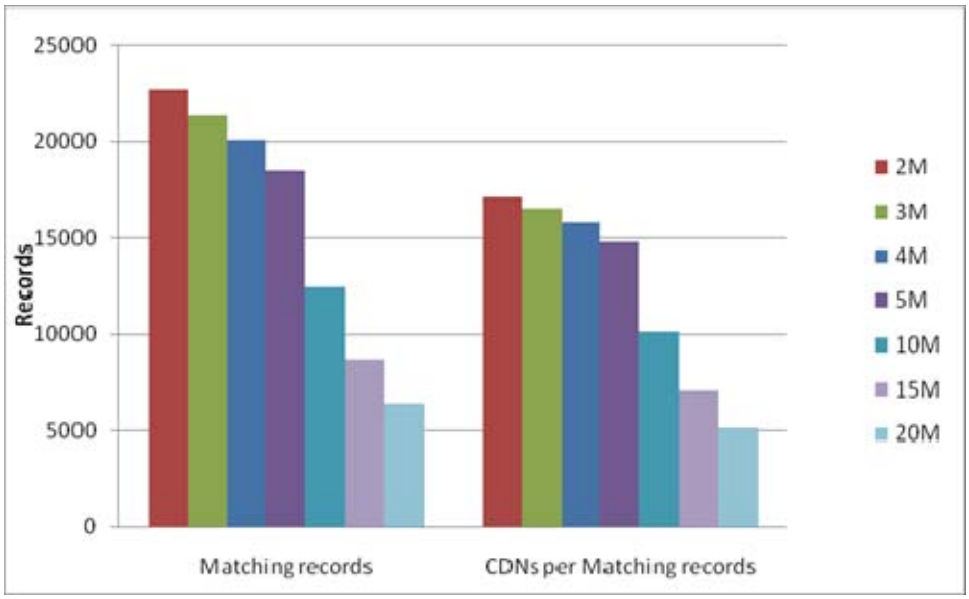

Figure 15. Matching records and CDNs per Matching records. 76,911 Total Records in sample.

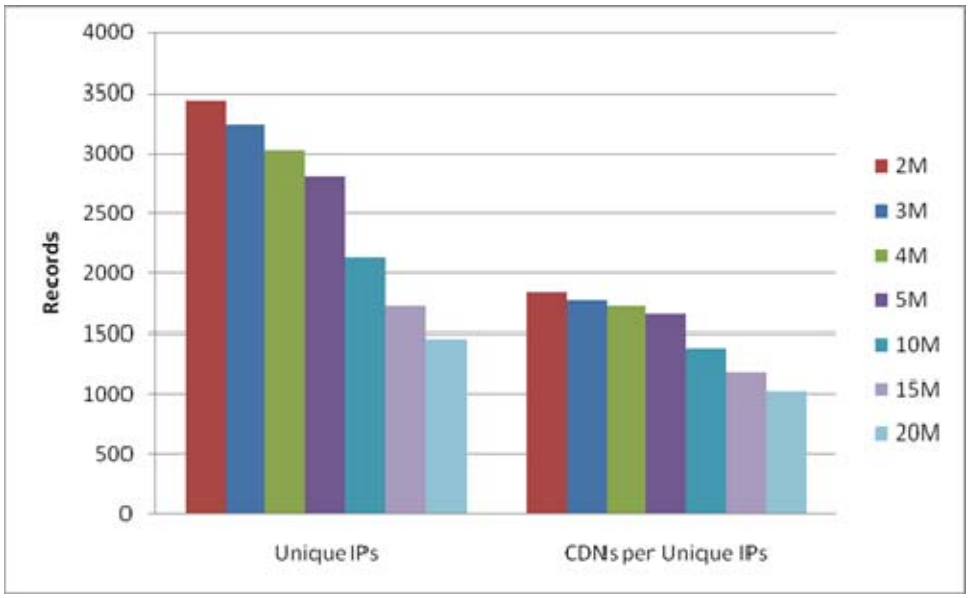

Figure 16. Unique IPs and CDNs per Unique IPs.

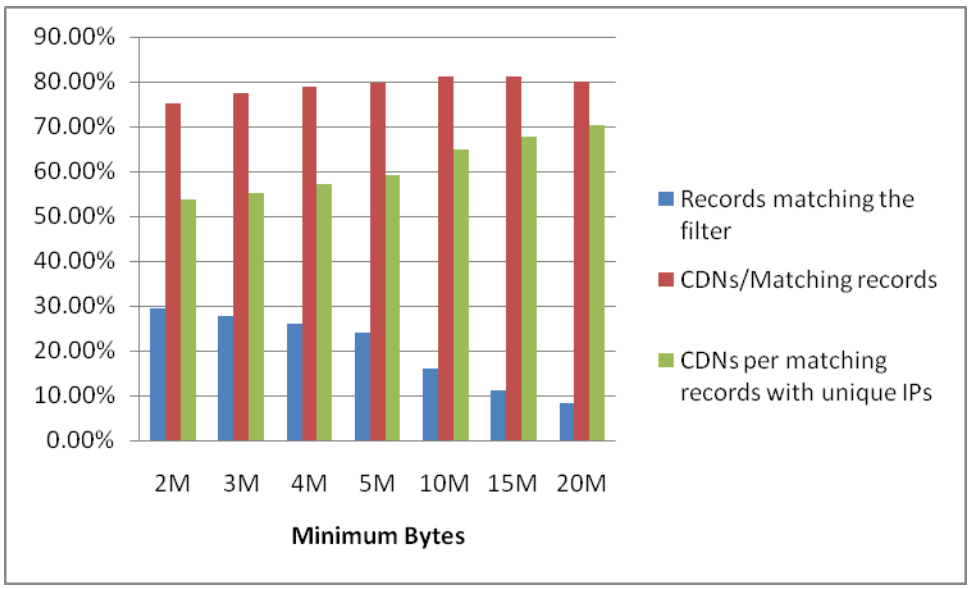

Figure 17. Percent matching vs Minimum session size

E. TRENDS

To confirm our streaming video predictions, we queried Centaur and retrieved 22 hours of flow records, in two-hour intervals, from August 13 thru August 14, 2010. We will call this dataset A13. We used "silk_Parser.pl" to parse the data and return statistics. With the exception of performing reverse DNS lookups, we followed the same procedure used to examine A12. Figure 18 depicts the percentage of CDNs per two-hour interval, using several minimum session sizes, as well as the overall average CDN detection percentage. Figure 19 depicts overall percentage of streaming traffic volume. We made several observations while examining Figure 18 and Figure 19.

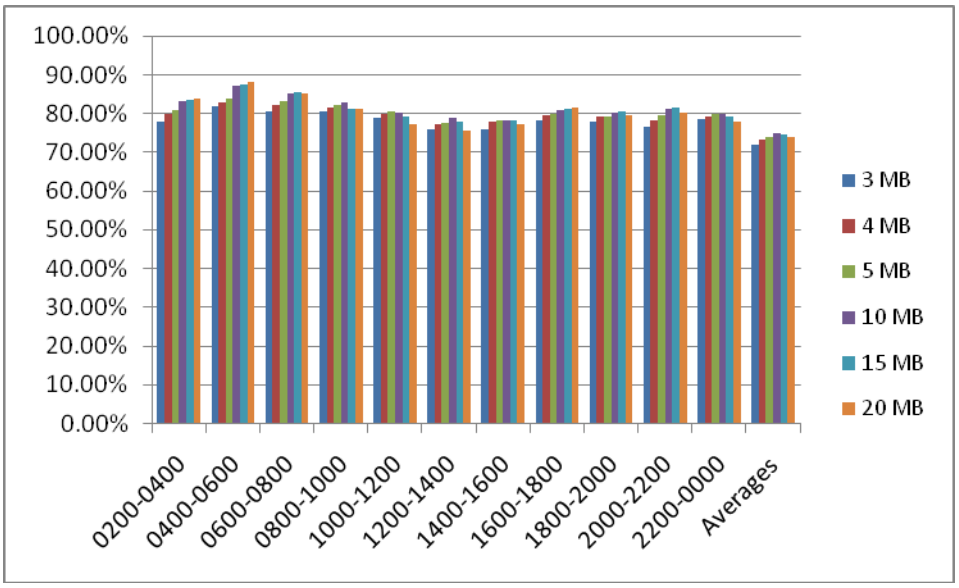

Figure 18. Percent of flows classified as CDN with six different minimum session size thresholds (3, 4, 5, 10, 15, and 20 MB)

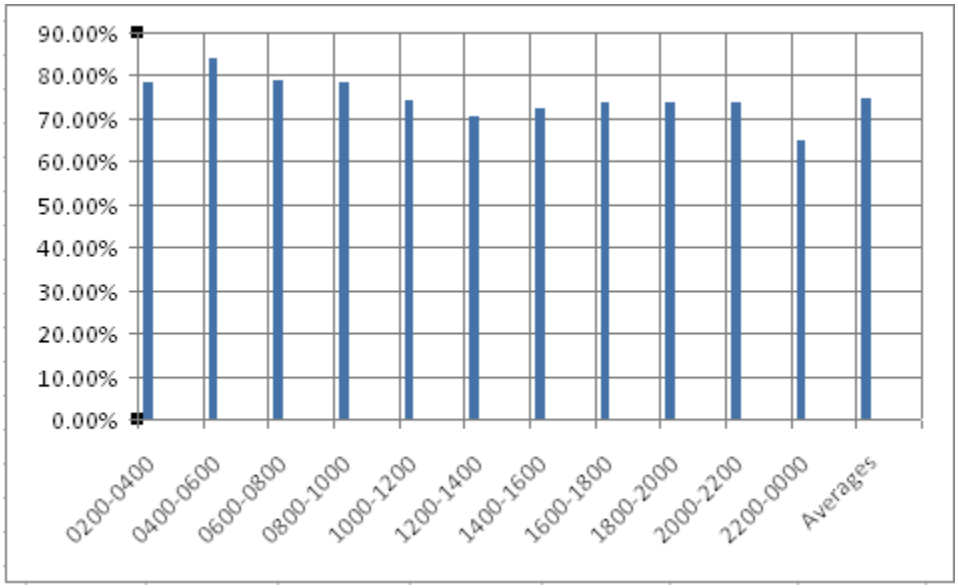

Figure 19. Percent of traffic volume classified as CDN with six different minimum session size thresholds (3, 4, 5, 10, 15, and 20 MB)

The first observation was that a minimum session size of 10 MB produced the highest percentage of CDNs detected across the 22-hour period. This is the same as the results from the A12 dataset. As previously discussed, this

represents two potential issues. Either the majority of streaming video observed on the DISA network is greater than 10 MB in size, or we have failed to identify several CDNs that offer streaming videos with a size less than this threshold. If the minimum session size were raised to 10 MB, there would be fewer false positives as indicated by the increase in percentage of flows associated with CDNs. The side effect would be that any video less than 10 MB would go undetected. If the filter remains below 10 MB, the probability of CDN detection decreases.

We also observed that CDN detection varied during certain times of the day. We converted the GMT to EDT and observed that the highest percentage of detection occurred from 0800 to 1000 EDT. This corresponds to the start of the workday for the majority of the DoD on the Eastern U.S. Seaboard. The percentage decreased, leading up to the lowest period of CDN detection from 1800 to 2000 EDT. This decrease correlates to the end of the workday and the evening meal. This analysis of times and the predicated activities are speculative and based on EDT. For a more accurate determination, we would need to separate traffic flows into regions and examine each with respect to the local time zone.

F. SUMMARY

This chapter verified our algorithm is capable of predicting streaming video at greater than 80% probability. The probability may be higher, but without complete listings of CDNs or Deep Packet Inspection, we have no way to validate this claim. We also chose to limit flow collection to sessions less than 200 MB due to limitations inherent in the workstations used to collect and analyze the data, as indicated in Chapter IV. We are confident that our algorithm would produce similar results against larger session sizes due to the profiles examined in Chapter IV, which large file downloads and streaming video made up all of traffic with a session size above 200 MB. Unless the server throttles the connection during a file transfer, streaming content is transmitted at a much

slower rate. This is mainly due to the limited cache available for the streaming video player to store the media. We determined that a minimum byte size per session of 10 MB predicted the greatest percentage of CDNs in the captured flows. We discussed the benefits and drawbacks of increasing the minimum session size. Additional research is required to fine-tune the filter criteria as well as a more refined list of CDNs to verify results. Based on our findings, we are confident our methods will identify streaming content on the DoD networks, as well as provide an area of future research that may impact bandwidth considerations.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION AND RECOMMENDATIONS

The goal of this thesis was to analyze and detect behavioral differences between streaming video utilizing HTTP and other types of Internet traffic. Technological limitations restrict an analyst from querying massive datasets; our study provides an iterative approach towards this problem. Our research focused on algorithms, various network analysis tools, and heuristic methods to establish definitive patterns for detecting streaming video from the information available in flow sessions.

This study revealed full packet captures were not feasible due to resources required to store, process, and analyze large amounts of data. Flow data is the standard used to generate traffic statistics in a network. Due to the limited information available in a flow trace, our keyword parsing technique and the IO graphs generated from the packet captures are not possible. We also determined that IP addresses could not be used to trace streaming media back to the root provider due to presence of CDNs. In the initial attempt to analyze flow data, we examined the ratio of upload to download traffic during a flow session. This resulting disparity in transfer sizes is indicative of streaming video or large file transfers; however, this technique would not allow differentiation between the two.

Bit rates were plotted from tested samples and we noticed file transfers had a higher bit rate than streaming content. We documented that the streaming videos in our testing datasets had a bit rate between .3 Mbps and 1.2Mbps. Our second parameter was a 45 second duration time for filtering advertising content. The final parameters used were minimum and maximum session sizes. The lower and upper asymptotic bounds were set to 3 MB and 200 MB respectively. These parameters were applied to filter the session data collected from Centaur. Analysis of 30 hours of session data resulted in an 80% prediction of CDN streaming content.

A. STUDY OVERVIEW

This analysis provided exploration of current Department of Defense policies, processes, and procedures designed to oversee network management, as well as identify and analyze content delivery technologies used by both industry and government organizations.

CDNs use a variety of methods to deliver content including, but not limited to, manual asset copying, active web caches, and global hardware load balancers.

This thesis should serve as a springboard for deeper research aimed at analyzing much larger datasets using our documented testing methods. In addition, further research on Content Distribution Network's address space is necessary to better bound the domain of possible consolidated distribution points. This will assist in identifying sources that provide streaming video. CDN technology is difficult to assess because root providers contract with multiple CDNs to stream their content. This makes tracing an IP to the root provider problematic.

In addition, we studied the Abacus algorithm specifically developed to identify anomalies in Peer-to-Peer networks. This algorithm allowed for identifying and classifying streaming video over P2P networks due to an individual application's UDP profile. Our analysis revealed streaming video uses a disproportionate amount of the allotted bandwidth for DoD NIPRNet traffic. We surmise this amount of streaming video will continue to increase as multimedia content continues to become more prevalent.

We tested multiple techniques to classify streaming video on captured data from known streaming video servers, as well as servers that offered sizeable file downloads. Although signatures were very similar, the relative

transfer rate for a file transfer was higher than that for a streaming video, given the same network infrastructure. This disparity allowed us to develop our own algorithm to predict streaming video.

Our analysis characterizes some distinct behaviors of HTTP streaming media by using session size and duration for modeling specific behaviors. This analysis allowed us to study some new and interesting techniques that hold promise if incorporated into firewall guidelines and policies. Such policies are necessary for establishing filters for IP address ranges, as well as developing strategies for restricting inappropriate activity. Information provided herein should serve as a major step in providing DISA with grounded analysis for detecting traffic patterns based on Centaur flow data. Some aspects hold sufficient promise to warrant further research.

B. FUTURE WORK

During this research, there were some areas uncovered that require future research work.

- The prevalence of Content Distribution Networks and the challenge of validating their streaming service after classification. What are the operating guidelines between a client and a CDN provider? Do CDNs favor a specific client base (i.e., pornography)? What is the address space of all the popular CDNs?
- Probability of detecting cyber attacks by examining network flow characteristics. What signature exists in flow data that can be exploited to assist analysts with detecting cyber activity?
- Larger dataset analysis for network behavior predictions with the goal of improving the probability of predicting streaming content to more than 80%. Further research with larger testing samples or larger file sizes could improve probability metrics.
- What is the bandwidth impact of streaming advertisements on DISA networks? During our research, we discovered that the majority of streaming content sessions captured were online advertisements.

C. RECOMMENDATIONS

Additional research in network anomalies using Centaur tools will require the researcher to have a working knowledge of the SiLK toolset to query, parse, and analyze massive databases. The researcher should also liaise with a DISA trusted agent early in the research process to ensure information is processed through all appropriate departments within DISA.

APPENDIX

A. KEYWORDS.PL

```
#!/usr/bin/perl
#===============================================================================
#
#         FILE:  keywords.pl
#
#         USAGE:  ./keywords.pl
#
#         DESCRIPTION: Reads TCP flowgraphs from the flowgraphs directory, performs a
#                              keyword search which is sent to STDOUT and counts unique IP
#                              addresses which are sent to keyword_scrip_IP_results.txt.
#         AUTHOR:  Mark Heller
#         CREATED:  08/19/2010
#===============================================================================

$| = 1;

use strict;
use warnings;

my (@array,@ip_output);
my $path = "../flowGraphs";
my $max_add = 0; #tracks the max number of ip addresses for output

#read all files in the directory into an array
opendir(DIR, $path);
while (my $file = readdir(DIR)){
        if ($file =~/^\w/){
                push (@array, $file);
        }
}
closedir DIR;

@array = sort {uc$a cmp uc($b)} @array;

#print column headers
printf ("%-25s%15s%15s%15s%15s%15s%15s%15s\n","Filename","HTTP
Count","akamai","video","player","porn","swf","flv","max frames");

#analyze each file in the array
my $num_graphs=0;  #counts the number of flow_graphs, used for 2d array
foreach my $file (@array){
        my
($http_count,$aka_count,$vid_count,$ply_count,$prn_count,$swf_count,$flv_count,$ack_count,$
previous)=(0)x9;
        my (@ip_array,@unique);
        open(IFILE,"$path/$file") or die "$!";
        my @lines = <IFILE>;
```

```
close IFILE;
foreach my $line (@lines){
	chomp($line);
	if ($line =~/ad/){
		next;
	}
	if ($line =~/HTTP: Continuation or non-HTTP traffic/){
		$http_count++;
	}
	if ($line =~/akamai/){
		$aka_count++;
	}
	if ($line =~/video/){
		$vid_count++;
	}
	if ($line =~/player/){
		$ply_count++;
	}
	if ($line =~/porn/){
		$prn_count++;
	}
	if ($line =~/swf/){
		$swf_count++;
	}
	if ($line =~/flv/){
		$flv_count++;
	}
	if ($line =~/Ack=/){
		my @word1 = split(/Ack=/,$line,2);
		my @word2 = split(/\s/,$word1[1],2);
		if($word2[0]-$previous >= 1440){
			$ack_count++;
		}
		$previous = $word2[0];
	}
	if ($line =~/(\d{1,3}\.){3}\d{1,3}/){
		my @word1 = split(/\s/,$line);
		foreach my $ip (@word1){
			#filter out private addresses
			if (($ip =~/^(\d{1,3}\.){3}\d{1,3}$/) && ($ip !~
			/(^127\.0\.0\.1)|(^10\.)|(^172\.1[6-9]\.)|(^172\.2[0-9]\.)|(^172\.3[0-
			1]\.)|(^192\.168\.)/)){
				push (@ip_array,$ip);

			}
		}
	}
	else{
		next;
	}
	my %hash = map { $_, 1 } @ip_array; #converts array into a hash table
	@unique = keys %hash; #removes any duplicate ip address
	@unique= map {s/\s+//g; $_} sort map {s/(\d+)/sprintf “%3s”, $1/eg; $_} @unique;
```

```
        }
        printf (“%-25s%15d%15d%15d%15d%15d%15d%15d\n”,$file,$http_count,
        $aka_count,$vid_count,$ply_count,$prn_count,$swf_count,$flv_count,$ack_count);
        $ip_output[$num_graphs][0]=$file;  #store filename in position [0] of array.
        my $num_ip=1; #counts number of IP address per flow_graph.
        foreach my $ip (@unique){
                $ip_output[$num_graphs][$num_ip]=$ip;
                $num_ip++;
        }
        if ($num_ip > $max_add){
                $max_add = $num_ip;
        }
$num_graphs++;
}
open (OUT, “>../results/keyword_scrip_IP_results.txt”) or die “$! error trying to openfile”;
my $row=0;
while($row < $max_add){
        my $col = 0;
        while($col < @ip_output){
                if($ip_output[$col][$row]){
                        printf OUT (“%20s”,$ip_output[$col][$row]);
                }
                else{
                        printf OUT (“%20s”,'');
                }
                $col++;
        }
        print OUT “\n”;
        $row++;
}
close OUT;
```

B. SILK_PARSER.PL

```
#!/usr/bin/perl
#===============================================================================
#
#         FILE:  silk_Parser.pl
#
#         USAGE:  ./silk_Parser.pl <input file> <output file>
#
#         DESCRIPTION: Reads SiLK statistics from the input file, filters based on bps, byte size,
#                      duration and source port.  The port is hard coded to 80/443.  Output is
#                      sent to STDOUT.  IP addresses are collected, checked for duplicates,
#                      known CDN servers and sent to ips_to_lookup.txt
#         AUTHOR:  Mark Heller
#         CREATED:  09/06/2010
#===============================================================================

$| = 1;

use strict;
use warnings;
use Net::IP::Match::Regexp qw( create_iprange_regexp match_ip);
my $infile = $ARGV[0];
my (@ip_array,@unique);
#===============================================================================
#USER SETTINGS
my $min_bytes=3000000;
my $max_bytes=200000000;
my $min_bps=300000;
my $max_bps=1200000;
my $min_dur=45;
my $max_dur=1800;
my $re2 = create_iprange_regexp({ '204.178.110.64/27' => 'Akamai Technologies'});
#===============================================================================
open FILE, $infile or die "$! error trying to openfile";
my @file = <FILE>;
close FILE;
my $cdn_per_uniq_count=0;
my $cdn_count=0;
my $rec_match=0;
my $ip_uniq = 0;
my $rec_total = 0;

open( OUT, ">silk_script_results.txt" )         or die "$! error trying to openfile";
open( IP,  ">silk_script_IP_addresses.txt" ) or die "$! error trying to openfile";
printf OUT ( "%15s|%5s|%15s|%5s|%10s|%10s|%8s|\n",
   "sIP", "sPort", "dIP", "dPort", "Packets", "Bytes", "Duration" );

foreach my $line (@file){
        chomp ($line);
        $line =~ s/\s*//g;
        if ($line !~/^\d/){
```

```
                next;
        }
        $rec_total++;
        my @var = split(/\|/,$line,8);
        my $sip = $var[0];
        my $sport = $var[1];
        my $packet = $var[4];
        my $byte = $var[5];
        my $dur = $var[6];
        #Remove any private addresses
        if ($sip =~ /(^127\.0\.0\.1)|(^10\.)|(^172\.1[6-9]\.)|(^172\.2[0-9]\.)|(^172\.3[0-
        1]\.)|(^192\.168\.)/){
                next;
        }
        #Remove any instances of 0 in the denominator
        if ($dur == 0){
                next;
        }
        my $bps = $byte*8/$dur;
        pop(@var);  #remove last element of the array
        if ($byte > $min_bytes && $byte <= $max_bytes && $bps > $min_bps && $bps <=
        $max_bps && $dur > $min_dur && $dur <= $max_dur && $sport =~/(80|443)/){
                push(@ip_array,$sip);
                $rec_match++;
                printf OUT ( "%15s|%5d|%15s|%5d|%10d|%10ddd|%8d|\n",
                        $var[0], $var[1], $var[2], $var[3], $var[4], $var[5], $var[6] );
        }


}
close OUT;
my %hash = map { $_, 1 } @ip_array; #converts array into a hash table
@unique = keys %hash; #removes any duplicate ip address
@unique= map {s/\s+//g; $_} sort map {s/(\d+)/sprintf "%3s", $1/eg; $_} @unique;

foreach my $line (@ip_array){
        if(match_ip($line,$re2)){
                $cdn_count++;
        }
}

foreach my $line (@unique){
        my $match=match_ip($line,$re2);
        if($match){
                $cdn_per_uniq_count++;
                printf IP ("%-15s  %s\n",$line,$match);
        }
        else{
                printf IP "$line\n";
        }
        $ip_uniq++;
}
close IP;
```

```
print “$min_bytes < Bytes <= $max_bytes  |  $min_bps < bps <= $max_bps  |  $min_dur <
Duration(secs) <= $max_dur\n”;
printf (“%-41s %d\n”,'Number of records:',$rec_total);
printf (“%-41s %d\n”,'Number of Matching records:',$rec_match);
printf (“%-41s %.*f%s\n”,'Precent of records matching our
filter:',2,100*$rec_match/$rec_total,'%');
printf (“%-41s %d\n”,'Number of CDNs:',$cdn_count);
printf (“%-41s %.*f%s\n”,'Precent of CDNs per Matching
records:',2,100*$cdn_count/$rec_match,'%');
printf (“%-41s %d\n”,'Number of Unique IPs:',$ip_uniq);
printf (“%-41s %d\n”,'Number of CDNs per Unique IPs:',$cdn_per_uniq_count);
printf (“%-41s %.*f%s\n”,'Precent of CDNs per Unique
IPs:',2,100*$cdn_per_uniq_count/$ip_uniq,'%');
```

C. REVERSEDNS.PL

```
#!/usr/bin/perl
#===============================================================================
#         FILE:  reverseDNS.pl
#
#         USAGE:  ./reverseDNS.pl
#
#         DESCRIPTION: Reads a list of ip addresses from the file ip_to_lookup.txt and performs a
#                              reverse DNS lookup.
#         NOTES:          Original version found at
#                              http://www.linuxquestions.org/questions/programming-9/
#                              fastest-way-in-perl-to-reverse-dns-lookup-14941/
#         AUTHOR:  Mark Heller
#         CREATED:  09/08/2010
#===============================================================================

use strict;
use warnings;

use constant TIMEOUT => 2;
$SIG{ALRM} = sub {die “timeout”};
my %CACHE;
open (IP, “../results/ip_to_lookup.txt”) or die “$! error trying to openfile”;
my @ip = <IP>;
close IP;
foreach my $line (@ip){
        chomp($line);
        print “$line\t”;
        my $dns = lookup($line);
        print “$dns\n”;
}


sub lookup {
   my $ip = shift;
   return $ip unless $ip=~/\d+\.\d+\.\d+\.\d+/;
   unless (exists $CACHE{$ip}) {
        my @h = eval <<'END';
        alarm(TIMEOUT);
        my @i = gethostbyaddr(pack('C4',split('\.',$ip)),2);
        alarm(0);
        @i;
END
                $CACHE{$ip} = $h[0] || undef;
        }
        return $CACHE{$ip} || '';
}
```

D. LIST OF KNOWN OR SUSPECTED CDN NETWORKS

Accelia, Inc	43.253.0.0/16		
Akamai Technologies	64.224.201.112/28	128.11.1.128/28	204.10.28.0/22
	64.224.201.128/28	128.11.1.144/30	204.178.110.32/27
	64.240.98.32/27	128.11.1.148/32	204.178.110.64/27
	65.126.84.0/24	128.11.10.235/32	204.8.48.0/22
	66.119.205.0/27	128.11.10.236/30	207.195.205.16/28
	66.152.103.64/26	128.11.10.240/29	209.170.115.0/24
	69.192.0.0/16	128.11.10.248/31	209.170.116.0/24
	69.22.137.0/24	128.11.10.250/32	209.170.117.0/24
	69.22.162.0/23	128.11.104.16/28	209.170.118.0/24
	69.22.164.0/24	128.11.28.16/28	209.170.94.0/24
	69.22.165.0/25	128.11.58.32/28	209.208.33.224/27
	69.27.160.0/20	173.222.0.0/15	209.221.135.128/27
	72.164.7.0/25	184.50.0.0/15	209.98.82.64/27
	72.246.0.0/15	184.84.0.0/14	216.127.199.224/28
	96.16.0.0/15	198.31.3.64/26	216.243.20.0/26
	96.6.0.0/15	198.77.126.64/29	216.246.122.0/24
	128.11.1.116/30	199.93.170.16/28	216.88.155.208/28
	128.11.1.120/29		
Amazon.com, Inc	72.21.192.0/19	205.251.192.0/18	207.171.160.0/19
	204.177.154.0/23	204.246.160.0/19	216.137.32.0/19
Beyond The Network	63.216.0.0/13	205.252.0.0/16	207.226.0.0/16
America, Inc	65.72.0.0/16	206.161.0.0/16	209.8.0.0/15
	205.177.0.0/16	207.176.0.0/17	
Carpathia Hosting, Inc	65.118.210.0/24	69.5.64.0/19	174.140.128.0/19
	66.117.32.0/19	173.245.96.0/19	209.222.128.0/19
	66.197.0.0/17		
CDNetworks Inc	4.59.182.64/26	174.35.0.0/17	208.80.248.0/22
	66.114.48.0/20		
Cogent Communications	64.17.48.0/20	66.250.0.0/16	209.17.96.0/20
	64.202.0.0/19	66.28.0.0/16	209.41.192.0/18
	64.254.192.0/19	66.71.224.0/20	216.168.64.0/20
	66.102.96.0/19	206.183.224.0/19	216.177.96.0/19
	66.102.96.0/19	207.230.0.0/19	216.229.128.0/20
	66.132.0.0/17	207.254.144.0/20	216.28.0.0/15
	66.213.165.0/24	209.115.0.0/17	216.55.80.0/20
	66.213.166.0/24	209.146.0.0/17	
Cognitive Networks, Inc	204.93.184.0/22		
Dailymotion S.A.	188.65.120.0/21	195.27.182.0/24	
Global Crossing	64.208.0.0/16	204.152.166.0/23	208.178.0.0/16
	64.209.0.0/17	204.245.0.0/18	208.48.0.0/18
	64.210.0.0/17	204.246.192.0/18	208.48.128.0/18
	64.211.0.0/17	206.132.192.0/18	208.48.192.0/20
	64.211.128.0/18	206.132.64.0/18	208.48.224.0/19
	64.211.192.0/19	206.165.0.0/16	208.49.0.0/16
	64.212.0.0/14	206.41.0.0/19	208.50.0.0/17
	64.76.0.0/16	206.57.0.0/17	208.50.192.0/18
	67.16.0.0/15	207.136.160.0/19	208.51.0.0/16
	146.82.0.0/16	207.138.0.0/16	209.130.128.0/18
	159.63.0.0/16	207.218.0.0/17	209.130.192.0/19
	162.97.0.0/16	207.218.128.0/18	
Google/YouTube	64.15.112.0/20	195.59.171.0/24	208.65.152.0/22
	74.125.0.0/16	208.117.224.0/19	213.146.171.0/24
	173.194.0.0/16		
Hurricane Electric, Inc	64.62.128.0/17	66.160.192.0/20	209.51.160.0/19
	64.71.128.0/18	66.220.0.0/19	216.218.128.0/17
	65.19.128.0/18	72.52.64.0/18	216.218.130.136/29

	65.49.0.0/17	74.82.0.0/18	216.66.0.0/18
	66.160.128.0/18	184.104.0.0/15	216.66.64.0/19
Level 3 Communications	24.56.96.0/20	199.183.192.0/18	207.221.64.0/18
	24.75.0.0/18	199.183.32.0/19	207.222.128.0/19
	24.75.128.0/20	199.183.64.0/18	207.222.176.0/20
	24.75.64.0/19	199.35.0.0/19	207.222.224.0/19
	24.75.96.0/20	199.35.128.0/17	207.222.64.0/18
	4.0.0.0/8	199.35.32.0/20	207.227.0.0/16
	8.0.0.0/8	199.35.96.0/19	207.7.0.0/18
	63.132.0.0/16	199.75.0.0/16	207.7.192.0/18
	63.133.0.0/17	199.76.0.0/15	207.83.0.0/16
	63.208.0.0/13	199.78.0.0/16	207.90.128.0/18
	64.140.0.0/18	199.92.0.0/14	207.92.144.0/20
	64.140.64.0/19	204.160.0.0/14	207.92.160.0/19
	64.152.0.0/13	204.164.0.0/14	207.92.192.0/20
	64.192.0.0/14	204.198.0.0/15	207.92.224.0/20
	64.200.0.0/16	204.30.0.0/19	207.92.48.0/20
	64.246.192.0/19	204.30.128.0/17	207.93.128.0/19
	64.30.0.0/18	204.30.48.0/20	207.93.160.0/20
	64.31.128.0/18	204.30.64.0/18	207.93.208.0/20
	64.66.64.0/19	204.31.16.0/20	207.93.32.0/19
	64.66.96.0/20	204.31.160.0/19	207.93.96.0/20
	64.8.0.0/18	204.31.224.0/19	207.94.144.0/20
	64.8.64.0/19	204.31.32.0/19	207.94.16.0/20
	64.9.0.0/17	204.31.64.0/20	207.94.176.0/20
	65.56.0.0/14	204.31.96.0/19	207.94.192.0/20
	65.77.0.0/16	204.32.0.0/17	207.94.224.0/19
	65.88.0.0/14	204.32.144.0/20	207.94.48.0/20
	66.114.192.0/18	204.32.160.0/19	207.94.80.0/20
	66.147.128.0/18	204.32.192.0/19	207.94.96.0/19
	66.147.192.0/19	204.33.0.0/18	207.95.0.0/19
	66.159.0.0/19	204.33.128.0/19	207.95.128.0/20
	66.170.128.0/20	204.33.176.0/20	207.95.160.0/20
	66.243.0.0/17	204.33.192.0/18	207.95.224.0/19
	66.243.128.0/18	204.33.96.0/19	207.95.48.0/20
	66.251.192.0/19	205.128.0.0/14	207.95.64.0/20
	66.51.48.0/20	205.180.0.0/14	207.95.96.0/20
	67.24.0.0/13	205.184.0.0/16	209.0.0.0/16
	67.63.0.0/19	205.187.128.0/19	209.100.0.0/16
	67.63.176.0/20	205.187.176.0/20	209.108.0.0/18
	67.72.0.0/14	205.187.192.0/18	209.108.128.0/19
	67.96.0.0/14	205.187.32.0/20	209.108.176.0/20
	67.97.181.0/24	205.187.80.0/20	209.108.192.0/19
	67.97.182.0/24	205.224.0.0/14	209.108.240.0/20
	69.44.0.0/15	206.15.0.0/19	209.108.64.0/19
	72.0.96.0/19	206.192.0.0/17	209.108.96.0/20
	72.236.0.0/15	206.215.0.0/20	209.109.0.0/19
	131.119.0.0/16	206.215.128.0/17	209.109.128.0/19
	131.192.0.0/16	206.215.32.0/19	209.109.176.0/20
	165.236.0.0/16	206.215.64.0/18	209.109.224.0/19
	166.90.0.0/16	206.216.112.0/20	209.109.32.0/20
	171.75.0.0/16	206.216.144.0/20	209.109.96.0/19
	192.156.170.0/23	206.216.16.0/20	209.110.128.0/20
	192.156.172.0/22	206.216.160.0/19	209.110.16.0/20
	192.156.176.0/21	206.216.192.0/18	209.110.160.0/19
	192.156.184.0/22	206.216.32.0/19	209.110.208.0/20
	192.156.188.0/23	206.216.64.0/19	209.110.224.0/19
	192.187.168.0/21	206.240.0.0/16	209.110.32.0/19
	192.187.176.0/20	206.241.0.0/16	209.110.96.0/19

	192.187.192.0/18	206.242.0.0/16	209.111.0.0/17
	192.2.0.0/16	206.243.0.0/16	209.111.144.0/20
	192.216.0.0/16	206.251.96.0/19	209.111.160.0/19
	192.221.0.0/16	206.32.0.0/14	209.111.192.0/19
	192.231.42.0/24	206.54.224.0/19	209.164.128.0/18
	192.233.0.0/16	207.112.128.0/17	209.224.0.0/16
	192.239.0.0/16	207.115.128.0/17	209.241.0.0/16
	192.31.48.0/24	207.120.0.0/14	209.244.0.0/14
	192.52.71.0/24	207.175.0.0/16	209.84.0.0/16
	192.80.92.0/22	207.220.128.0/18	216.127.224.0/19
	198.112.0.0/14	207.220.192.0/20	216.140.0.0/14
	198.31.0.0/16	207.220.224.0/19	216.158.160.0/20
	198.76.0.0/14	207.220.32.0/19	216.174.0.0/18
	198.92.0.0/14	207.220.80.0/20	216.202.0.0/16
	199.183.128.0/20	207.220.96.0/20	216.22.64.0/18
	199.183.16.0/20	207.221.128.0/17	216.248.0.0/18
	199.183.160.0/19	207.221.32.0/19	
Limelight Networks	68.142.64.0/18	69.164.0.0/18	208.111.128.0/18
	69.28.128.0/18	206.223.120.0/24	
Limelight Networks Asia Pacific	203.77.184.0/21		
Reflected Networks, Inc	64.210.128.0/19	208.99.64.0/19	216.18.160.0/19
	66.254.96.0/19	209.239.160.0/20	
SoftLayer Technologies	66.228.112.0/20	75.126.0.0/16	208.101.0.0/18
	67.228.0.0/16	173.192.0.0/15	208.43.0.0/16
	74.86.0.0/16	174.36.0.0/15	
XO Communications	64.0.0.0/14	204.238.120.0/24	209.116.0.0/14
	64.178.0.0/18	204.91.0.0/16	209.135.192.0/18
	64.178.64.0/19	205.158.0.0/16	209.19.192.0/18
	64.220.0.0/15	205.197.0.0/16	209.193.128.0/17
	64.244.0.0/15	206.111.0.0/16	209.21.128.0/17
	64.35.0.0/17	206.173.0.0/16	209.220.0.0/16
	64.48.0.0/16	206.181.0.0/16	209.31.0.0/16
	64.50.0.0/17	206.183.64.0/19	209.48.0.0/15
	64.55.0.0/16	206.196.64.0/19	209.68.192.0/18
	65.104.0.0/14	206.205.0.0/16	209.95.0.0/19
	65.44.0.0/14	206.225.32.0/19	216.0.0.0/14
	66.104.0.0/14	206.251.128.0/19	216.105.0.0/19
	66.2.0.0/15	206.251.128.0/19	216.112.0.0/16
	66.236.0.0/14	206.81.32.0/19	216.120.0.0/17
	66.88.0.0/15	206.83.64.0/19	216.149.0.0/16
	67.104.0.0/13	207.101.0.0/16	216.156.0.0/16
	67.152.0.0/14	207.110.0.0/18	216.203.128.0/17
	67.88.0.0/13	207.155.128.0/17	216.209.0.0/19
	71.4.0.0/15	207.158.128.0/18	216.22.128.0/17
	140.239.0.0/16	207.180.64.0/19	216.237.128.0/18
	165.117.0.0/16	207.208.0.0/16	216.250.64.0/19
	192.188.72.0/24	207.238.0.0/15	216.30.0.0/17
	192.188.72.0/24	207.70.64.0/18	216.30.128.0/20
	198.180.32.0/19	207.8.0.0/17	216.4.0.0/15
	198.187.32.0/19	207.86.0.0/15	216.50.0.0/16
	199.125.128.0/17	207.88.0.0/16	216.51.0.0/17
	199.34.32.0/19	208.176.0.0/15	216.55.0.0/18
	204.192.0.0/16	208.36.0.0/15	216.99.224.0/19
Yahoo! Inc	64.156.215.0/24	8.12.144.0/24	216.136.128.0/22
	64.157.4.0/24	98.136.0.0/14	216.136.172.0/22
	64.41.224.0/23	184.165.0.0/16	216.136.203.0/24
	64.58.76.0/22	204.71.200.0/22	216.136.204.0/24
	66.163.160.0/19	208.67.64.0/21	216.136.224.0/22

67.195.0.0/16	209.131.32.0/19	216.136.232.0/22
68.180.128.0/17	209.225.40.0/24	216.155.192.0/20
69.147.64.0/18	209.247.158.0/24	216.252.96.0/19
76.13.0.0/16	216.109.94.0/23	

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Adobe. *Adobe.* http://kb2.adobe.com/ (accessed September 6, 2010).

Buda, G., D. Choi, R. Graveman, and C. Kubic. "Security Standards for the Global Infomation Grid." *ieeexplore.* www.ieeexplore.ieee.org (accessed July 15, 2010).

CERT. "ISilk, A graphical front-end for the Silk Tools." *User's Guide.* Pittsburgh, PA: N/A, May 2009.

Chairman Joint Chiefs of Staff. *CJCSI 6211.02C.* Washington, July 9, 2008.

Chilton, Kevin P. Land War Net Speech [Transcript]. http://www.stratcom.mil/speeches/16/Land_War_Net_Speech (accessed October 1, 2010).

Collins, M. "YouTube Activity on NIPRNET From May, 2005 to December, 2006." *Publication NetSA 2007-09*, February 2007: 1–9.

Cormen, T., et al. *Introduction to Algorithms.* Boston, MA: McGraw-Hill, 2001.

Defense Information Systems Agency. *DISA.* January 1, 2010. www.disa.mil (accessed August 22, 2010).

DISA. "Surety, Reach, Speed." *DISA.* www.disa.mil (accessed August 15, 2010).

Dorobek, C. *DoD May Pull Key Net from the Internet.* August 26, 2002. http://www.insidedefense.com (accessed July 1, 2010).

Jones, K. *Real Digital Forensics Computer Security and Incident Response*. Upper Saddle River: Pearson Education. 2006.

Hubenko, V. P., Raines, R.A., Mills, R. F. Baldwin, R. O. Mullins, B. E. and Grimaila, M. R. "Improving the Global Information Grid's Performance through Satellite Communications Layer Enhancements." *IEEE Communications Magazine*, November 1, 2006.

Kohler, E, M., and Floyd, S. *Port Numbers.* January 15, 2010. www.iana.org (accessed September 6, 2010).

Krishnamurthy, B., and Rexford, J. *Web Protocols and Practice: HTTP/1.1 Networking Protocols, Caching, and Traffic Measurement.* New York, NY: Addison-Wesley, 2001.

National Security Agency. "GIG." *NSA.* February 00, 2008. www.nsa.gov (accessed August 22, 2010).

Nelson, C., and McAllister, S. *FIGHTCLUB: Leveraging the Analytic Research & Development Sandbox in the CND Community Data Center.* Technical, Ft Meade: National Security Agency, 2009.

Report to Congress: D. 110–77 (Senate Armed Services Committee, September 01, 2007).

Rossi, D., and Valenti, S. *Fine Grained Traffic Classification With Netflow Data.* Technical, INFRES, Telecom Paris Tech, France, Caen: IWCMC.

Rossi, D. *AbacusDemo.* http://perso.infres.enst.fr (accessed September 6, 2010).

Satterthwaite, C. "Space Surveillance and Early Warning Radars: Buried Treasure for the Information Grid." *DTIC.* June 2000. www.dtic.mil (accessed July 1, 2010).

U.S. Department of Defense. "DoD Directive 09-026." *DTIC.* February 25, 2010. www.dtic.mil (accessed July 23, 2010).

U.S. Department of Defense. *DoD Directive 81001.1.* Washington, DC, June 1, 2007.

U.S. Department of Defense. "USSTRATCOM." *JTF GNO WARNORD 07-003 IAP Al/Security Filter Update.* May 17, 2007. www.ussstratcom.mil (accessed August 1, 2010).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. DISA, Code PEO-IA22
   Arlington, Virginia